NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

INTERNETWORKING: IMPLEMENTATION OF MULTICASTING AND MBONE OVER FRAME RELAY NETWORKS

by

Ridvan Erdogan

September 1996

Thesis Advisors: Don Brutzman
Michael Zyda

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1996	Master's Thesis

4. TITLE AND SUBTITLE
INTERNETWORKING: IMPLEMENTATION OF MULTICASTING AND MBONE OVER FRAME RELAY NETWORKS

5. FUNDING NUMBERS

6. AUTHOR
Erdogan, Ridvan

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/ MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/ MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT

The major problems addressed by this thesis research are how to implement multicast over the Monterey BayNet to enable live audio/video for distance learning, how to safely integrate regional Frame Relay multicast with the global MBone, and how to monitor multicast connectivity over the Monterey BayNet. To implement multicast and MBone over the Monterey BayNet without using dedicated multicast servers, we enabled Protocol Independent Multicast (PIM) protocol on already-installed Frame-Relay-capable routers. By implementing multicast over Monterey BayNet, we show that the current MBone software provides the same performance that it provides on regular Internet connections even on low-speed (128Kbps) Frame Relay network connections and low-cost personal computers. In order to control the scope of the regional multicast and to safely integrate regional Frame Relay multicast with the global MBone, we used administratively controlled multicast group addresses (224.0.1.20) in addition to the use of time-to-live (TTL) control mechnanism. This eliminates global duplication of multicast packet delivery. Public-domain multicast monitoring tools are used to monitor the multicast connectivity through internetworks. Since these tools are available only to UNIX-based platforms, they cannot be used by the regional sites that mostly have Windows and Macintosh platforms. We developed Web-accessible multicast monitoring pages in order to meet the multicast monitoring needs of the regional sites. Participating sites are now able to monitor regional multicast connectivity by accessing these pages, which permits remote problem diagnosis. That was previously impossible. Finally we synopsize firewall requirements for secure and effective use of multicast.

14. SUBJECT TERMS
Multicast, Multicast Backbone (MBone), Frame Relay, Wide-Area Network (WAN), Network Monitoring

15. NUMBER OF PAGES
148

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

Approved for public release; distribution is unlimited

INTERNETWORKING: IMPLEMENTATION OF MULTICASTING AND MBONE OVER FRAME RELAY NETWORKS

Ridvan Erdogan
Lieutenant Junior Grade, Turkish Navy
B.S., Turkish Naval Academy, 1989

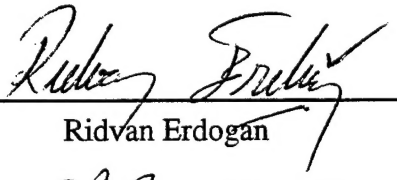
Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

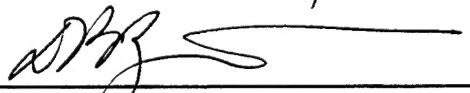
from the

NAVAL POSTGRADUATE SCHOOL

September 1996

Author: 

Ridvan Erdogan

Approved By: 

Don Brutzman, Thesis Advisor

Michael Zyda, Thesis Advisor

Ted Lewis, Chair,
Department of Computer Science

ABSTRACT

The major problems addressed by this thesis research are how to implement multicast over the Monterey BayNet to enable live audio/video for distance learning, how to safely integrate regional Frame Relay multicast with the global MBone, and how to monitor multicast connectivity over the Monterey BayNet.

To implement multicast and MBone over the Monterey BayNet without using dedicated multicast servers, we enabled Protocol Independent Multicast (PIM) protocol on already-installed Frame-Relay-capable routers. By implementing multicast over Monterey BayNet, we show that the current MBone software provides the same performance that it provides on regular Internet connections even on low-speed (128 Kbps) Frame Relay network connections and low-cost personal computers. In order to control the scope of the regional multicast and to safely integrate regional Frame Relay multicast with the global MBone, we used administratively controlled multicast group address (224.0.1.20) in addition to the use of time-to-live (TTL) control mechnanism. This eliminates global duplication of multicast packet delivery.

Public-domain multicast monitoring tools are used to monitor the multicast connectivity through internetworks. Since these tools are available only to UNIX-based platforms, they cannot be used by the regional sites that mostly have Windows and Macintosh platforms. We developed Web-accessible multicast monitoring pages in order to meet the multicast monitoring needs of the regional sites. Participating sites are now able to monitor regional multicast connectivity by accessing these pages, which permits remote problem diagnosis. That was previously impossible. Finally we synopsize firewall requirements for secure and effective use of multicast.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to thank my best friend, my wife, Nurten, for all her love, support, and patience. Without her, this thesis would never have happened. Working with a student whose native language is not English must be very difficult. I also would like to thank Dr. Don Brutzman and Dr. Michael Zyda for their patience, support, and understanding. Also many thanks to David Stihler from the Monterey County Office of Education. Without his help, we never would have had a chance to complete this work.

I. INTRODUCTION

A. BACKGROUND

The Monterey BayNet is a wide-area network (WAN) which connects students, educators, schools and research institutions throughout the Monterey and Santa Cruz counties. It is the end product of joint efforts of independent organizations and volunteer groups. The project was initially designed by the Monterey Bay Regional Education Futures (MB ReEF) group. It focuses on the K-12 community and provides full Internet access to its sites in a cost-effective way. The Initiative for Information Infrastructure and Linkage Applications (I^3LA) network design team has been instrumental in the design and implementation of the Monterey BayNet. End-user needs and design goals were determined and the necessary technology was selected to meet these needs and goals.

Connectivity for this project has been funded by a California Research and Education Network (CalREN) grant. CalREN was created by Pacific Bell (PacBell) in 1993 to fund projects focusing on education, health care, community, government and commercial business areas. CalREN funds project connectivity for a maximum of two years. Frame Relay, ISDN and ATM are the data communication technologies available from PacBell.

Since Frame Relay offered greater access speeds, savings and a clear transition path for increased bandwidth, it has been selected as the wide-area network (WAN) connectivity service for the Monterey BayNet (Bigelow, 1995). Frame Relay is a connection-oriented, Layer 2, WAN protocol. Protocols such as IP and IPX can be encapsulated into Frame Relay frames. Internet Protocol (IP) is the only protocol that is allowed over the Monterey BayNet. IP packets are transmitted over the network by using the multiprotocol encapsulation feature of Frame Relay. The Cisco 2500 family of routers has been selected for use on the Monterey BayNet. These routers offer a range of technological options to end user sites. The current releases of Cisco ISO, or router operating system, have built-in multicast support. This enables native IP multicast packet delivery across the networks. The focus of this thesis is on the specifics of implementing IP multicast over Frame Relay regionally.

B. MOTIVATION

An Internet compatible videoconferencing capability has been assessed as an end-user requirement for local schools by the net design team. Videoconferencing can be used for distance learning, which can improve and diversify the quality of network-based education. Distance learning has the potential for dramatic improvements in training and education. A variety of different technologies are used for distance learning. However, most comprehensive versions do not scale well, are inflexible, and (most importantly) are expensive. Since low cost is a precondition of the Monterey BayNet project, an inexpensive, easy-to-implement solution needs to be found for distance learning.

The Multicast Backbone (MBone) is a technology that can be used for distance learning. MBone permits many-to-many communications over the Internet. It provides live audio and video to its users today. Current studies show that distance learning using the MBone is a feasible approach (Emswiler, 1995).

When Monterey BayNet was initially implemented, the MBone was still in its infancy. This technology was available only to high-power workstations, while Monterey BayNet school sites had either Windows or Macintosh platforms. Therefore implementation of network-based videoconferencing was delayed.

1996 was a glorious year for the MBone. After built-in multicast support of Windows 95 operating system was announced, the MBone-related software became available for Windows platforms. MBone tools for Windows platforms work just as well as they do on UNIX platforms.

MBone was originally designed to provide multicast (many-to-many communication) functionality to internetworks. Multicasting is now a standard part of the TCP/IP protocol suite. Most router vendors (including Cisco) have announced multicast support in their products. Cisco routers are able to provide native multicast packet delivery across internetworks. Since Cisco routers are used by Monterey BayNet sites and IP is allowed within the Monterey BayNet Frame Relay sites, it is now possible to implement multicasting over the network and use MBone for distance learning purposes.

C. PROBLEM SUMMARY

Frame Relay is the link-layer protocol which provides WAN connectivity to the Monterey BayNet. Frame Relay multicast service specifications rely on special multicast servers because Frame Relay is a connection-oriented protocol while multicast is a connectionless service. However, no multicast servers were available for use in the Monterey BayNet. Pacific Bell, the Frame Relay service provider for the Monterey BayNet, does not provide multicast services. This thesis mainly focuses on how multicasting can be deployed over the Monterey BayNet without using dedicated multicast servers. Our motivation is to enable live audio/video using the MBone tools for distance learning.

D. THESIS ORGANIZATION

The next chapter surveys related work in the area of Frame Relay wide-area networking, multicasting and MBone. It discusses the use of MBone for distance learning purposes, the cost-effective design of MBone classrooms for distance learning, and the monitoring of networks.

Chapter III precisely defines the questions that this thesis research tries to answer:

- Can multicasting and MBone be implemented over the Monterey BayNet, which is a Frame Relay WAN?
- How can continuous multicast traffic be sustained over the Monterey BayNet?
- How can the regional Frame Relay multicast be safely merged with the global MBone?

Chapter IV defines multicasting and Multicast Backbone (MBone). The underlying network concepts of multicasting and MBone are discussed. It provides background information for those new to multicasting and MBone.

Chapter V describes how IP multicasting is implemented over a Frame Relay WAN, Monterey BayNet. The Frame Relay protocol itself and native Frame Relay multicasting are introduced first. Since Frame Relay is a connection-oriented protocol, native Frame Relay multicasting is hard to support. By using the technology that the Monterey BayNet sites have, it is possible to implement IP multicasting and take advantage of the functionality that MBone provides. The site needs for implementing multicasting, the required configuration changes, and

the installation of MBone software for Windows platforms are documented in this chapter.

Chapter VI presents three multicast monitoring tools that can be used to monitor multicast connectivity: Automated Mrouter Checking Program (AMCP), *mrinfo* Gateway, and *mtrace* Gateway. The continuity of multicast traffic is an important issue when dealing with the delivery of video and audio for videoconferencing purposes. An automated monitoring tool, AMCP, is developed to monitor the NPS local MBone. It monitors the multicast connectivity on an hourly basis and documents the results on a world-wide accessible Web page. Public domain multicast monitoring tools (such as *mrinfo* and *mtrace*) are available to UNIX and UNIX-like platforms (such as Linux, FreeBSD) only. However, most Monterey BayNet sites are schools with Windows or Macintosh platforms. WWW and scripting languages (i.e, *CGI/perl*) permit the execution of programs via Web pages. *mrinfo* and *mtrace* Gateways meet the monitoring needs of Monterey BayNet sites by using the functionality provided by WWW and scripting languages.

Chapter VII documents the experimental results. There are very few existing implementation examples of multicasting over Frame Relay networks. The performance of MBone over Frame Relay connections and on low-cost personal computers (PCs) is tested and documented in this chapter. The tests were conducted during the experimental MBone sessions created for test purposes.

Chapter VIII contains conclusions. It reviews what this thesis research tried to achieve. We concluded that multicasting and MBone are possible over Frame Relay networks, and that the existing technology that Monterey BayNet sites have is sufficient for the implementation of multicasting over the Monterey BayNet. This final chapter closes with recommendations for future work.

II. RELATED WORK

A. INTRODUCTION

This chapter discusses related work that complements this thesis. Many of the issues discussed in this thesis require re-examination of underlying technologies. Each complementary work is discussed briefly.

B. INTERNETWORKING: PLANNING AND IMPLEMENTING A WIDE-AREA NETWORK (WAN) FOR K-12 SCHOOLS (BIGELOW, 1995)

We are living in an information age. Fast information exchange between individuals is essential. Today, people are willing to get information whenever and whereever they need it. The Monterey BayNet has been designed and implemented to meet this growing need. It is a wide-area network (WAN) which connects kindergarten through twelfth grade (K-12) students, educators and research institutions throughout Monterey and Santa Cruz counties on the central California coast. In his thesis research (Bigelow, 1995) R. Jon Bigelow gives the details for planning, designing and implementing a Frame Relay WAN. Lessons learned demonstrate how technical and human challenges are solved. This thesis research also provides information for students and teachers who want to better understand the technology behind the Internet.

The on-line version of this thesis can be found at

http://www.stl.nps.navy.mil/~rjbigelow/thesis/toc.html

C. MULTICAST ROUTING IN A DATAGRAM INTERNETWORK (DEERING, 1991)

Multicast is a many-to-many communication method that permits sources to send a single copy of a data packet to a group address that causes the data packet to be delivered to multiple recipients. For shared-medium networks where all hosts listen to the same media, such as Ethernet and FDDI, multicast is easy to support. Complexity arises when the multicast services are extended to internetworks. A multicast service model which is applicable to internetworks has been devised by Steve Deering. In his doctoral dissertation (Deering, 1991), he introduced a new

service model for multicasting in datagram internetworks and a set of new store-and-forward multicast routing algorithms to support that service model. Multicast extensions for internetworks makes the existing Internet services more efficient and robust, allowing existing datagram networks to support applications which require real-time delivery of data.

The postscript version of this dissertation is available at

ftp://gregorio.stanford.edu/vmtp-ip/sdthesis.part1.ps.Z Part 1
ftp://gregorio.stanford.edu/vmtp-ip/sdthesis.part2.ps.Z Part 2
ftp://gregorio.stanford.edu/vmtp-ip/sdthesis.part3.ps.Z Part 3

D. FRAME RELAY PVC MULTICAST SERVICE AND PROTOCOL DESCRIPTION (SWALLOW, 1994)

Frame Relay is a connection-oriented link-layer protocol. Because of the connection-oriented nature of Frame Relay protocol, the implementation of multicast (which is typically connectionless) requires deployment of special multicast servers within the Frame Relay cloud. (Swallow, 1994) is the Multicast Service Implementation Agreement for Frame Relay which describes Frame Relay multicast services. Unlike general IP multicast, Frame Relay multicast is connection oriented and relies on the dedicated multicast servers. Since multicast is an optional service for Frame Relay networks, establishing multicast services is specified as an administrative operation that requires coordination between a participating service provider and service users.

The on-line version of this document is available at

http://frame-relay.indiana.edu/5000/Approved/FRF.7/FRF7-TOC.html

E. MBONE PROVIDES LIVE AUDIO AND VIDEO ACROSS THE INTERNET (MACEDONIA AND BRUTZMAN, 1994)

In this paper, Mike Macedonia and Don Brutzman describe the underlying network concepts of the Multicast Backbone (MBone) and give general guidance about it. MBone is one of the most interesting applications of the Internet. It is a virtual network which is layered on top of the portions of the physical Internet. It is called a virtual network because it uses the same physical media that the Internet does. It is designed to provide multicast functionality to internetworks. The Mbone was first utilized in March when live audio was multicast from an

IETF Meeting. Adequate processing power and built-in audio capability of today's workstations and deployment of IP multicasting has accelerated the use of MBone. Today MBone is used for live audio and video transmission across the Internet.

This paper is available in hypertext, postscript, and plain text formats at

ftp://taurus.cs.nps.navy.mil/pub/i3la/mbone.html	Hypertext Form
ftp://taurus.cs.nps.navy.mil/pub/i3la/mbone.ps	Postscript Form
ftp://taurus.cs.nps.navy.mil/pub/i3la/mbone.txt	Text Form

F. MBONE: INTERACTIVE MULTIMEDIA ON THE INTERNET (KUMAR, 1996)

(Kumar, 1996) is the first book written exclusively about the MBone. Vinay Kumar gives readers the chance to explore the Internet and MBone resources. The details of how multicasting works in an internetwork environment, how MBone began and how it has grown are provided. The existing MBone software tools are also introduced and discussed in detail. The book also provides information to system administrators who want to use MBone more effectively. The table of contents of this book can be found at

http://www.best.com/~prince/techinfo/mbonetoc.txt

G. AN ANALYSIS OF INTERNET'S MBONE: A MEDIA CHOICE PERSPECTIVE (GAMBRINO, 1994)

In this case study, John R. Gambrino examines the perceived effectiveness of the MBone and analyzes its capabilities and limitations. He compares the MBone versus face-to-face viewer perceptions of the different communication media by using data gathered during an experiment between NPS and the Monterey Bay Aquarium Research Institute's (MBARI). This case study showed that MBone capabilities as of 1994 were more effective for reducing uncertainty than resolving equivocal communication situations. Improvements in frame rate and resolution since that time have shown significant qualitative improvements in Internet-based videoconferencing. Further analytic study along the lines of this thesis appear warranted.

H. INTERNETWORKING: WORLD-WIDE MULTICAST OF THE HAMMING LECTURES FOR DISTANCE LEARNING (EMSWILER, 1995)

In this case study, Tracey L. Emswiler investigates distance learning combined with MBone. She shows that MBone is an economically feasible approach that works. Distance learning has the potential for realizing dramatic improvements in training and education. Distance learning technology can provide education to individuals who want it, when and where they need it. Televised instruction, traditional videoconferencing and some other technologies have been used for distance learning. Televised instruction uses the conventional broadcast or cable technology. The cost and the lack of interaction makes this approach infeasible in most cases. Traditional videoconferencing technologies are equipment dependent, expensive and inflexible. They do not scale to a large number of participants or to participants at many sites. In this respect, MBone can provide interesting opportunities for distance learning. Dr. Richard Hamming's course "Learning to Learn" was transmitted world wide over the MBone, three times weekly for an entire quarter. This case study demonstrated that the MBone was able sustain an ongoing event. A user manual for the MBone tools is also provided.

I. INTERNETWORKING: MULTICAST AND ATM NETWORK PREREQUISITES FOR DISTANCE LEARNING (TAMER, 1996)

The Internet, the World Wide Web, and the Multicast Backbone (MBone) have been used in a variety of ways for distance learning. However, it is not clear how practical they are as an affordable alternative to proprietary commercial videoconferencing systems. Video TeleConferencing (VTC) classrooms have obvious value and utility but they are limited to communicate with only a small number of similar VTC facilities. In his Master's Thesis (Tamer, 1996), Murat Tamer tries to determine the specific benefits and drawbacks of Internet technologies in support of distance learning. This thesis research shows that an MBone classroom is significantly less expensive than a VTC room. Furthermore, many schools have the minimal equipment needed for a Web/MBone classroom in their inventory: Windows 95 or Linux personal computer with audio capability (video card optional). Consequently, many schools can afford Internet-based distance learning even though they cannot afford VTC rooms.

This thesis research is available at

http://www.stl.nps.navy.mil/~iirg/tamer/thesis.html

J. INTERNETWORKING: AUTOMATED LOCAL AND GLOBAL NETWORK MONITORING (EDWARDS, 1996)

To monitor a network means to have the capacity to determine the route taken by the transmission, display the time that it required, determine what percentage (if any), of the transmission was lost, and most importantly determine what is wrong with the network. The capability of monitoring large networks and internetworks is not available today. The inability to monitor a network can exist in equal measure in a local area network, but the complexity of the problem increases with each remote link of the connection. Monitoring capabilities do exist but with significant hindrances. Commercial tools are prohibitively expensive while public domain tools are cryptic. Neither represent a viable option for many network users. With proper automation and integration, however, public domain tools can deliver accurate and timely information on network status and performance. Evan Edwards documents in his Master's Thesis (Edwards, 1996) how public domain tools can be used for automated monitoring of networks. He also provides scripts used for automation and monitoring. This work is closely related to the multicast network monitoring tools developed in Chapter VI and Appendix E of this thesis. This thesis research is available at

http://www.stl.nps.navy.mil/~iirg/edwards/thesis.html

K. SUMMARY

People learn more from what they see and hear. This interaction greatly impacts the effectiveness of education. Extending multicast functionality to the Monterey BayNet and using MBone for educational purposes will enable distance learning capabilities in support of regional education efforts. The works listed in this section provide a broad set of background references for understanding what this thesis is trying to achieve.

III. PROBLEM STATEMENT

A. INTRODUCTION

This chapter defines the problems addressed by this thesis research. Recently it has been shown that internetworked distance learning using multicast audio and video can have a positive impact on education and training (Gambrino, 1994). Also it has been shown that the Multicast Backbone (MBone) is a feasible approach that supports distance learning (Emswiler, 1995) (Tamer, 1996). This thesis mainly focuses on how multicasting can be implemented over the Frame Relay Monterey BayNet.

B. THESIS PROBLEMS EXAMINED

The Monterey BayNet is a wide-area network (WAN) designed and implemented to cost-effectively and fully connect educators, students and researchers to the Internet. The fundamental problem examined in this thesis is how to safely implement multicasting and Multicast Backbone (MBone) over the Frame Relay Monterey BayNet in order to take full advantage of existing Internet technologies.

The Monterey BayNet intends to provide the Monterey Bay region's educational system with access to new, up-to-date technology. One of the design goals was to implement full access to the Internet via videoconferencing. Videoconferencing across the Internet is possible using a number of different technologies. However, most of them are expensive, hard-to-use and not scalable. Current studies have shown that the MBone is the most cost-effective solution for videoconferencing. When Monterey BayNet was designed and implemented, MBone was in an early phase and MBone-related software was not yet available for Windows or Macintosh platforms. MBone was relying on multicast routers (mrouters) which are UNIX based workstations with a special software for multicast routing running on them, and logical connections set up between them. Today's advanced, rapidly changing technology make MBone possible even for PC platforms. Current routers also have built-in multicast support.

The technology appears to be available but the following question remains: "how can multicasting and MBone be implemented over the Monterey BayNet, which is a Frame Relay

WAN?"

Frame Relay is the technology for WAN connectivity that has been selected for the Monterey BayNet. It is a connection-oriented protocol, and as in other connection-oriented protocols (such as ATM), native multicasting is problematic. According to the Frame Relay specification (Swallow, 1994) multicasting ordinarily relies on special multicast servers. Pacific Bell, the Frame Relay service provider for Monterey BayNet, does not offer such a service. Therefore a multicast service model that is independent of servers and service providers needs to be implemented for the Monterey BayNet.

A further question is whether Monterey BayNet sites have equipment sufficient to support the multicast service model, or else what the site equipment needs are in order to support such a model.

Implementation of multicasting and MBone over Monterey raises a couple of additional questions: "what are the MBone and multicasting" and "what do they provide?" Multicasting and MBone are new technologies. They are both experimental. Without understanding the underlying concepts of multicasting and MBone, success cannot be expected when implementing multicasting and MBone for Frame Relay. Background information about multicasting and MBone is therefore provided in this thesis for current and future users of the Monterey BayNet.

Implementation of multicasting establishes a kind of virtual network. Monitoring of this virtual network becomes an important issue for sustainability. For continuous multicast traffic availability, the components of this virtual network need to be regularly monitored. Multicast monitoring tools therefore must be available for use on the Monterey BayNet sites. However, the commercial solutions are extremely expensive. This thesis provides effective, interactive and free software monitoring tools to ensure that multicast traffic problems can be diagnosed and corrected.

C. SUMMARY

Videoconferencing capability has been determined as an end-user requirement and a design goal of the Monterey BayNet during the design phase. However, due to the large scale of the Monterey BayNet project, this goal was originally delayed. That goal is now achieved by this thesis. Specifically, this thesis addresses three research questions:

- Can multicasting and MBone be implemented over the Monterey BayNet which is a Frame Relay WAN?
- How can continuous multicast traffic be sustained over the Monterey BayNet?
- How can the regional Frame Relay multicast be safely merged with the global MBone?

IV. MULTICAST SERVICES IN AN INTERNETWORK ENVIRONMENT

A. INTRODUCTION

Today's internetworks are predominantly based on point-to-point data communications. Most data communication occurs between two specific hosts and does not interfere with the remaining hosts on a network. However, there are times when a host wants to send a data packet to every other host or a set of hosts on the network. To meet such a need, different communication methods such as broadcasting (one-to-many communication) and multicasting (one-to-many or many-to-many communication) are defined in addition to unicasting (point-to-point communication).

Multicasting is an Internet Protocol (IP) service that permits sources to send a single copy of a data packet to a group address that causes the data packet to be delivered to multiple recipients. Multicasting is more efficient than requiring sources to send individual copies of a message to each recipient. For shared-medium networks where all recipients can hear all transmissions (such as Ethernet), multicasting is easy to support and has been used for many years. However, extending multicasting to internetworks causes some difficulties.

This chapter documents why multicasting is important for network applications and what benefits can be obtained by using it. The multicast service model first devised for datagram internetworks by Steve Deering is briefly discussed. The underlying concepts of IP multicasting are examined. Finally the Multicast Backbone (MBone) examined, perhaps the most interesting capability of the Internet. It provides multicast functionality to the Internet. The main objective in this chapter is to provide essential background technical information to those readers who are new to multicasting and MBone. This background knowledge is needed to understand subsequent thesis chapters.

B. UNICAST, BROADCAST, MULTICAST

Exchanging information between computers and human users is the fundamental purpose of data communication. In (ANSI, 1990) data is defined as "a representation of facts, concepts or instructions in a formalized manner, suitable for communication, interpretation, or processing by

human beings or by automatic means." Data can be identified; data can be described; data does not necessarily represent something physical in terms of the measurable world; but data can be used, namely to produce information (Stalling, 1991). Again in (ANSI, 1990), information is defined as "the meaning that is currently assigned to data by means of the conventions applied to data." Information is born when data is interpreted (Stalling, 1991). Thus exchanging information requires access to elements of data and the ability to transmit them.

In the simplest form, data communications occur between two devices that are directly connected by some form of point-to-point transmission medium. However, connecting two devices directly (point-to-point) is not practical because many devices are far apart and dedicated lines are expensive. Some devices require a link to many destinations at various times. Again, it is impractical to provide a dedicated wire between each pair of devices. Instead, each device is attached to a communication network, containing a collection of devices that wish to communicate.

In an Internet Protocol (IP)-based communication network, one of the following three communication methods to exchange information can be used: unicast, broadcast, and multicast communications. Each method is examined in detail.

Unicast is a single-destination, point-to-point communication method. A unicast transfer occurs when one sending host sends data to a single receiving host (Figure 4.1). The data is delivered only to a single destination, with intermediate hosts forwarding and/or discarding the message. The most common example of unicast communication is e-mail delivery. Only the recipient host and intermediate routers pay attention to the unicast packet.

Broadcast is a one-to-many communication method. Sometimes broadcast is needed to send data to all hosts in the communication network, such as an administrative shutdown warning sent to all hosts in an Ethernet Local-Area Network (LAN). Instead of sending a copy of the same data to each host, only one copy of the data is sent and each host in the LAN is expected to receive it. As seen in Figure 4.1, the broadcast message sent by Host A is received by all hosts on the LAN. Since broadcast packets might propagate and flood an internetwork, they are generally prohibited from passing across routers and are thus restricted to the confines of a LAN.

Multicast communication combines the strengths of unicast and broadcast communications. Like broadcast communication, only one copy of the data is sent and it touches every host in the communication network. Unlike broadcast, the data is processed only by the

hosts that intend to receive it. In other words, the multicast packets are sent to some self-selected subset of possible destinations or a group of hosts. As demonstrated in Figure 4.1, the multicast packets sent by Host D are received only by Host B and Host E that are willing to receive them. Since indiscriminate propagation of multicast packets might easily saturate an entire communication network, they are also prohibited from traversing routers unless special routing mechanisms are employed.

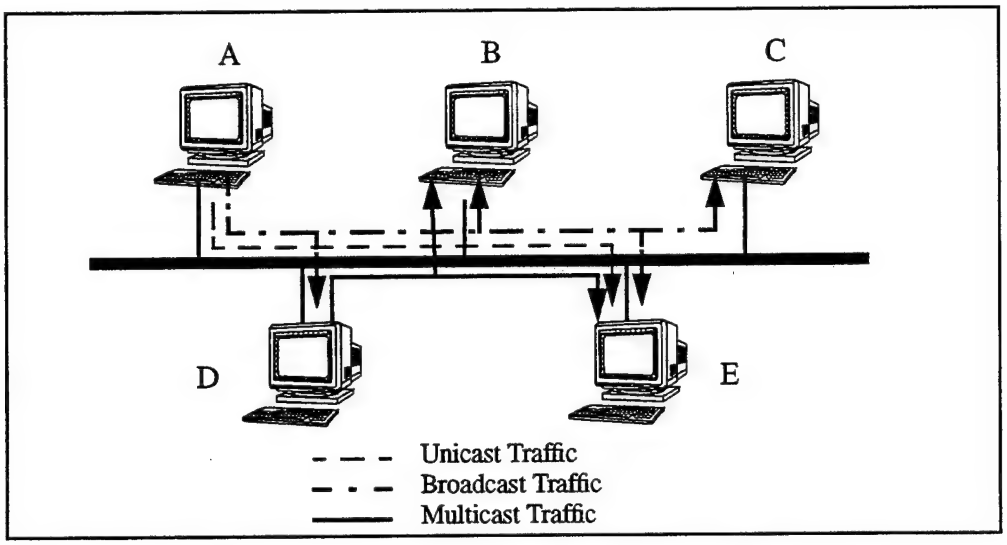


Figure 4.1 Unicast, Broadcast, and Multicast Communications within a Single LAN

Unicast and broadcast can be considered special forms of multicast. In unicast, the group of hosts has only one member and in broadcast, all hosts in the network are the members of that group. Thus, proper deployment and use of multicast has fundamental importance.

C. WHY MULTICAST?

Multicast packets can be easily distinguished from unicast and broadcast packets by the address header. Unlike unicast, individual multicast packets can be read by every host workstation on a LAN. Since redundant duplicated-data delivery is eliminated, bandwidth requirements are minimized. Multicast packets are screened by the hosts and unwanted packets are discarded. This is done at the hardware level and eliminates unnecessary computational burden at the application level.

Since multicast is widely used in distributed applications, it is an important facility for network applications. Multicast provides two important benefits to network applications.

The first major benefit is the sending of data packets to a group of hosts. It is not a good idea to send separate copies of the same data to each host. The sending of an e-mail message to multiple recipients does not cause a big problem. However, when sending high-bandwidth streams such as video or graphics, it is an expensive process. In this sense, multicasting is more efficient than unicasting. Multicast can reduce the transmission overhead on senders and receivers. Multicast can also reduce the overhead across the network and the time taken for all destinations to receive the information.

The second major benefit of multicast is the ability to reach a group of hosts whose individual addresses are unknown to the sender and whose membership may change over time. Identifying this set of destinations by a single multicast group address rather than by a list of individual addresses permits reaching destinations whose individual addresses are unknown or changeable. This use of multicast, called logical addressing or location-independent addressing, serves as a robust and simple alternative to membership configuration files, directory servers or other binding mechanisms. For instance, sensor readings can be continuously delivered to a self-selected, changeable set of monitoring stations.

LANs such as Ethernet and FDDI provide widely used multicasting services because of these benefits. Extending multicast services across an internetwork provides additional advantages. Existing multicast-based applications can be easily migrated to an internetwork environment without rewriting them to use multiple unicast streams or special-purpose servers. Existing internet services become more efficient and robust under higher loads. Finally, applications which require real-time data delivery, such as teleconferencing and audio/video distribution, can be supported by existing datagram networks.

D. MULTICASTING IN AN INTERNETWORK ENVIRONMENT

Multicasting on shared medium networks (such as Ethernet and FDDI) where all hosts share a common transmission channel is simple. The cost of multicasting those networks is the same as unicasting. Simply, IP datagrams are specified with a destination address that is a multicast address. This address is converted to a corresponding hardware address, such as an Ethernet address, and the packet is sent. The hosts willing to receive this packet notify their network interface cards that they want to receive datagrams destined for that multicast address.

The datagrams with that multicast address are caught at the hardware level and passed up to higher levels. This is similar to the process applied to unicast IP datagrams and does not cause any extra burden over the network.

However, complications arise when multicasting is extended beyond a single physical network and multicast packets pass through routers. Multicast packets are sent to a group address which identifies a set of destination group rather than individual destinations. Hosts intended to receive these packets may join and leave these groups at any time. Therefore, new models and protocols are needed to deliver only requested multicast packets from one physical network to another, and to determine if any hosts on a given physical network belong to a given multicast group. In particular, routing loops must be avoided.

Multicast was first introduced to internetworks by Steve Deering. In (Deering, 1991) methods for a LAN-like multicast service in an internetwork environment are provided. In the following subsections, the basics of IP multicasting are discussed.

1. Host Group Model

As discussed in the previous section, multicast packets are sent to a group of hosts rather than individual hosts. For datagram internetworks, a new multicast service model (called Host Group Model) was designed (Deering, 1991). Under this model, the set of destinations of a multicast packet is called a host group. It is identified by a single group address or multicast address. The destination address field of the IP datagram headers is used for identifying groups of hosts rather than individual hosts. Membership in a host group is dynamic. Hosts can join and leave the groups at will and need not coordinate with other group members or the potential senders to the group. This is important for applications such as videoconferencing where participants may come and go independently. The Host Group Model allows any host, whether or not that host belongs to the group, to send a packet to any group. The members of a host group may be spread across the internetwork. Host Group Model does not impose topological restrictions on group membership. This makes a collection of subnetworks appear like a single logical network.

Under this model, multicast groups can be either permanent or transient. A permanent group has a well-known, administratively assigned group address. It is the address of the group

which is permanent, not the membership to that group. Permanent groups may have any number of members at any one time, including zero. Transient groups exist only for as long as they have at least one member. They are created when they are needed and discarded when the number of members reaches zero. Temporary group addresses are used for transient groups.

2. Host Group Addressing (Multicast Addressing)

Each host group has a unique group address which can also be called a multicast address. The destination address field in the IP datagram header is used to specify multicast delivery. A single IP unicast address is statically bound to a single local network interface on a single IP network. Unlike the IP unicast address, an IP multicast address or host group address is dynamically bound to a set of local network interfaces on a set of internetworks. The IP host group address is not bound to a set of participating IP unicast addresses. The multicast routers do not need to maintain a list of individual members of each group. What multicast routers need to know is whether there are any hosts willing to receive that multicast packet.(Deering, 1991)

Class D IP addresses are reserved for the use of IP multicast. Class D addresses are distinguished from other IP address classes such as A, B, and C by looking at the four high-order bits. The four high-order bits of an IP multicast address is 1110. When expressed in dotted decimal notation, IP multicast addresses range from 224.0.0.0 through 239.255.255.255.

Some IP multicast addresses are assigned as well-known addresses by the Internet Assigned Numbers Authority (IANA) (Reynolds, 1994). These addresses are used to identify permanent host groups and reserved for the use of routing protocols, low-level topology discovery or maintenance protocols, and other special projects. This is similar in concept to the well-known TCP and UDP port numbers. IP multicast addresses other than the well-known multicast addresses are available for use by transient groups. Appendix A contains a full list of current IANA IP Multicast Addresses.

3. Protocols

When multicasting is extended across internetworks, new protocols are required to distribute multicast packets from one subnet to another, to determine which host groups are

present on which subnet, and to detect any host group which appears or disappears on a subnet. Currently, support for IP multicasting comes from three protocols: Internet Group Management Protocol (IGMP), Distance Vector Multicast Routing Protocol (DVMRP), and Protocol Independent Multicast (PIM) Protocol.

a. Internet Group Management Protocol (IGMP)

Internet Group Management Protocol (IGMP) is a protocol used by hosts and routers. It informs all the systems on a physical network which hosts currently belong to which multicast groups. Hosts running multicast applications use IGMP to join host groups. Routers use the same protocol to determine the location of hosts that have joined a multicast group and therefore know which multicast datagrams to forward onto which interfaces.

IGMP is based on a simple query-response scheme. Routers on which IGMP is enabled periodically send IGMP host-query messages to discover which host groups have members on their directly attached local subnetworks. Hosts respond to this query by generating host membership reports and list the multicast groups that hosts want to join. When the routers receive a packet addressed to a multicast group, the packet is forwarded on those interfaces that have hosts belonging to that group. The absence of host membership reports for a host group is meant to inform the router that the group is no longer present. If, after several queries, the router determines that there are not any local hosts that are members of a particular multicast group on a particular interface, the router stops forwarding packets for that group and sends a prune message to upstream routers which tells upstream routers not to forward multicast packets for that group.

IGMP is considered part of the IP layer. IGMP messages are transmitted in IP datagrams and specified with a protocol value of 2. Unlike other protocols, IGMP has a fixed size message with no optional data.(Stevens, 1994) (Deering, 1989) (Fenner, 1996)

b. Distance-Vector Multicast Routing Protocol (DVMRP)

DVMRP is a distance-vector routing protocol designed to support the forwarding of multicast datagrams through an internetwork. DVMRP is concerned with computing the shortest path back to a source (Semeria, 1996).

In a DVMRP router, all interfaces (physical or logical tunnel) are associated with a metric value that specifies the cost for a given port and a TTL threshold value that limits the scope of a multicast transmission. A multicast datagram is forwarded across an interface only if the TTL field on the packets IP header is greater than the TTL threshold assigned to the interface. DVMRP routers maintain a (source, group) pair for each multicast group. When a multicast datagram for a (source, group) pair is received for the first time, it is forwarded on all interfaces which meet the TTL threshold requirement if it arrives via a link that might be used to reach the source of the datagram. If the datagram arrives via any other link or does not meet the TTL requirement, it is discarded. The initial datagram is delivered to all leaf routers. The leaf routers send a prune message back towards the source if there are not any group join requests on their directly attached leaf subnetworks.(Semeria, 1996) This allows information about the absence of group members to be propagated back towards the source. Subsequent packets from the same source to the same group are blocked from travelling down unnecessary (i.e, pruned) branches. Protocol details can be found in (Partridge, 1988) and (Deering, 1991).

DVMRP uses IGMP to determine the absence and presence of the groups, as well as the location of hosts that have joined a multicast group. DVMRP control messages are encapsulated in IGMP messages. The type field of the IGMP message format is used to distinguish DVMRP messages from other control messages. The IGMP messages carrying DVMRP messages have a type value of 3 in their type fields.

c. Protocol Independent Multicast (PIM) Protocol

Protocol Independent Multicast (PIM) is an IP multicast protocol which is currently under development by the Inter-Domain Multicast Routing (IDMR) Working Group of Internet Engineering Task Force (IETF). Unlike DVMRP, PIM does not rely on any specific unicast routing protocol. It is designed to work with all existing unicast routing protocols. Unicast routing protocols are only needed to provide routing table information and adapt to topology changes.

PIM has two modes that allow it to work effectively with two different types of multicast traffic distribution patterns: dense and sparse modes.

Dense-mode PIM is designed for the following conditions.

1. Senders and receivers are in close proximity to one another
2. There are few senders and many receivers
3. The volume of multicast traffic is high (Cisco, 1996)

In dense-mode PIM, it is assumed that all routers are willing to forward multicast packets for a group. When a multicast packet is received by a dense-mode PIM router, it is simply forwarded on all downstream interfaces, except the interface on which the packet was received until explicit prune messages are received. If the router receives a multicast packet and there are not any directly connected members for that group or PIM neighbors, a prune message is sent back to the source. Details of protocol description can be found in (Jacobson, 1996).

Sparse-mode PIM is designed for the environments where

1. Senders and receivers are separated by wide-area network (WAN) links
2. Many multipoint data streams go to a relatively small number of LAN segments.

The sparse-mode PIM router assumes other routers do not want to forward multicast packets for a group unless there is an explicit request for that group. In sparse-mode PIM, one of the routers is designated as a Rendezvous Point (RP). The RP is responsible of collecting information about senders and making this information available to potential receivers. When a sender wants to send data, it first sends the data to the rendezvous point. When a receiver wants to receive data, it registers with the rendezvous point. (Deering, 1995) has the details of the protocol description.

Like DVMRP, PIM also uses IGMP to determine the location of hosts that have joined a multicast group. PIM control messages (for both dense and sparse modes) are encoded in IGMP messages. A value of 4 in the type field of a IGMP message format determines that it is a router PIM message. (Jacobson, 1996)

E. MBONE: THE VIRTUAL NETWORK

Communication is essential in our daily lives. The growing need for intuitive real-time human communication through computer networks cannot be ignored. The Multicast Backbone

(MBone) was created in part to meet this need. At the March 1992 Internet Engineering Task Force (IETF) meeting, IP multicasting for the Internet was first employed. From that meeting, live audio was transmitted for the first time and some 30 people participated remotely from Australia, Sweden, UK and the U.S., both listening and asking questions. The built-in audio capabilities of today's workstations and the deployment of IP multicast has accelerated the increase in the use of MBone (Macedonia, 1994). It has grown from 40 subnets in four different countries in 1992 to more than 2800 subnets in over 25 countries at the time of this writing (Kumar, 1996). Today it allows anyone to send live audio and video across the Internet.

The primary goal of MBone is to provide multicast functionality. It is a virtual network which is layered on top of sections of the physical Internet. In the MBone architecture, multicast-capable subnetworks such as Ethernet LANs (called islands) are connected to each other via logical links (called "tunnels") to support multicasting across the Internet. The tunnel endpoints are typically workstation-class machines that have operating system support for IP multicast and run the multicast routing daemon "*mrouted*", making them multicast routers (called mrouters). Since many of the existing routers are not capable of multicasting, the use of mrouters and tunnels remains necessary. Distance Vector Multicast Routing Protocol, DVMRP (Deering 1991), is the routing protocol implemented by the *mrouted* program.

Encapsulation is the process of placing an IP datagram inside a network packet or frame so that it can be sent across an underlying network. To be able to pass multicast traffic through the non-multicast-capable parts of the Internet, IP multicast packets are encapsulated as IP-over-IP, so that they look like normal unicast packets to intervening routers and subnetworks.

When an upstream mrouter receives an IP-multicast packet, it prepends another IP header and sets its destination address in the new header to be the unicast address of the multicast router at the end of the tunnel. The downstream mrouter at the end of the tunnel receives the packet, strips off the encapsulating IP header and forwards the previously embedded multicast packet to local hosts (and possibly other downstream mrouters) as appropriate. Figure 4.2 is a simple illustration of this process.

Multicast packets forwarded by mrouters can touch all workstations on the local network. It uses the same bandwidth whether it is received by one workstation or all workstations on the network. If such a stream jumps from network to network without any controls, it is possible to saturate the entire Internet very quickly. Therefore a mechanism is needed for limiting the scope

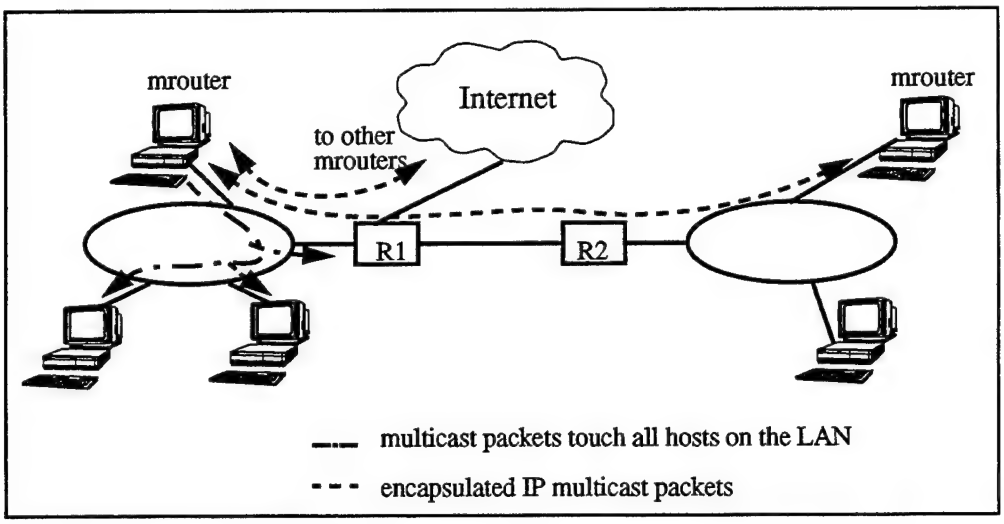


Figure 4.2 IP Multicast Packets are Encapsulated as IP-over-IP to Bypass the Non-multicast-capable Routers

of a multicast stream. The scope of a multicast stream can be controlled by using either truncated tunnels or pruned tunnels.

In a truncated multicast scheme, the lifetime of the multicast packets is limited. The time-to-live (TTL) field of standard IP header is used to determine the lifetime of the multicast packet. A specific TTL value is assigned by the client for each multicast packet sent to the network. Each tunnel has a threshold value. The threshold is the minimum TTL value that a multicast packet needs to be forwarded onto a given tunnel. For each *mrouted* that the multicast packets pass, the TTL is decremented by 1. If the packet's remaining TTL is lower than the threshold value of the tunnel that *mrouted* intends to use, the packet is dropped. By convention, an initial TTL value of 1 is used to limit the scope of the multicasting to local subnetwork. An initial TTL value of 32 is used to limit the scope to an organizational LAN (such as the NPS campus). An initial TTL value of 64 is used to limit the scope to the same geographic region. An initial TTL value of 128 is used to limit the scope to the same continent, and an initial TTL value of 255 is used to send multicast packets to the entire Internet. (Semeria, 1996)

In a pruning multicast scheme, multicast packets are not forwarded to sites or mrouter nodes unless they have expressed an explicit interest in packets via IGMP messages belonging to that specific multicast group address, and are within the TTL range of the sender. If a multicast packet for which it has no receiving clients or tunnels to forward it to is received by an mrouter, that packet is dropped by the mrouter. The mrouter also sends a signal to upstream mrouters that says it does not want packets with that address. The upstream mrouters that get this signal stop

sending packets that way. The pruned multicast scheme results in less unwanted multicast traffic to and from a subnet which in turn can reduce the possibility of congestion. Therefore pruned multicasting is sometimes called true multicasting. The older versions of *mrouted* software support only truncated multicasting. The later versions (version 3.x and later) support the pruning feature. The latest version of *mrouted* is 3.8 at the time of this writing.

Each tunnel has also a metric value. The metric value is used for routing and specifies a routing cost that is used in DVMRP. Mrouters use the metric value to compute the shortest path to the destination mrouters. MBone administrators can use metrics to designate whether routes are preferred routes or backup routes.

The beauty of the MBone is that it allows live audio and video distribution across the Internet (Macedonia, 1994). It often carries the audio and video multicasts of NASA Space Shuttle Missions, U.S. House and Senate Sessions, conferences and lectures. It is also used for distance learning purposes. Recent studies such as (Emswiler, 1995) have shown that distance learning can be combined with MBone to provide a feasible approach that works.The MBone software tools in Figure 4.3 make this possible. These tools allow MBone to be used for interactive multimedia over the Internet (Kumar, 1996).

F. SUMMARY

Multicast communication provides efficient multi-destination and robust unknown-destination delivery capabilities to network applications. It has been used by shared medium networks for years. Extending multicast capability to internetworks makes existing internet services more efficient, more robust and allows existing datagram networks to support applications that require real-time delivery.

The model and protocols developed by Steve Deering allow multicast services on internetworks and provide a LAN-like multicast service in an internetwork environment.

The Multicast Backbone (MBone) is one of the most interesting features of the Internet. It is called a virtual network because it shares the same physical media as the Internet. Networks of mrouters are used by MBone to support multicast across the Internet. It allows live audio/video delivery across the Internet. A variety of public domain software tools are available that run on most computer architectures.

***session directory (sd)*:** This is a tool used to allocate, reserve, and advertise multicast sessions. *sd* is the TV guide of the MBone (Kumar, 1996). It allows users to reserve and allocate multicast channels for distribution of media. It also allows users to examine and join these channels.
sdr: *sdr* is based on *sd*. It is also designed to allow advertisement and joining of multicast conferences on the MBone. *sd* and *sdr* are not compatible although gateways have been written.
***Multimedia Conference Control (mmcc)*:** This is rendezvous software. It is used to invite other participants into an MBone session. MMCC allows multiple sites to participate in interactive multimedia sessions, either on a person-to-person basis or in a multiparty scenario.(Kumar, 1996)
***Net Video (nv)*:** *nv* is a widely used MBone application tool. It allows real-time delivery of video across the Internet MBone.
***VideoConference (vic)*:** *vic* is the video tool which is primarily designed for multiparty conferencing applications. It has a flexible, and extensible architecture to support heterogenous environments and configurations. A variety of video encoding are available including H.261.
***Inria Videoconferencing System (ivs)*:** *ivs* is a tool that allows both video and audio data over the Internet. It is especially popular with European MBone users.
***Visual Audio Tool (vat)*:** *vat* is an audio tool and allows two or more Internet hosts to participate in voice-based conferencing.
Robust-Audio Tool (rat): *rat* is another audio tool for MBone. It uses the Real Time Protocol (RTP) Version 2 (Casner, 1996). It has a packet-loss protection mode (using redundancy) which provides enhanced performance during conditions of packet loss through the use of Forward Error Correction (FEC).
***Network Voice Terminal (NeVot)*:** NeVot also allows multiple users to participate in the real-time audio-based conferencing.
***WhiteBoard (wb)*:** *wb* allows multiple sites to share documents and drawings in real-time. Basically it is a document sharing program that operates in both multicast and unicast mode. *wb* works over the Internet in low-bandwidth conditions and provides reliable updates.
***Image Multicast Client (IMM)*:** It is a low-bandwidth application that allows graphic image distribution over the Internet. A client/server model is used for IMM. The server multicasts graphic images and the client receives and displays them on a user's desktop screen.

Figure 4.3 Widely used MBone tools

V. WIDE-AREA NETWORK (WAN) MULTICASTING OVER FRAME RELAY

A. INTRODUCTION

The Monterey BayNet is a wide-area network (WAN) which connects K-12 schools, libraries, and higher educational institutions throughout Monterey and Santa Cruz counties. It enables students and teachers to access environmental information and resources regionally via the Internet. Frame Relay has been selected as the WAN technology for the Monterey BayNet. Frame Relay is a connection-oriented, Layer 2 (Link Layer) protocol. As in other connection-oriented protocols (such as ATM), multicasting is problematic for Frame Relay. Frame Relay multicasting ordinarily relies on multicast servers within the network. However, most Frame Relay service providers do not have such servers and multicasting is not available to Frame Relay users.

This chapter documents how multicasting was implemented over the Monterey BayNet Frame Relay WAN. After a brief introduction to Frame Relay protocol itself, the following questions are answered: "what is Frame Relay" and "how does Frame Relay works?" Frame Relay multicasting is discussed next. Since a multicast server is not available to Monterey BayNet users, IP multicasting over Frame Relay was implemented using existing routers. Frame Relay permits multiprotocol encapsulation. The Internet Protocol (IP), the only traffic allowed to be routed within the Monterey BayNet, is one of the protocols that can be encapsulated within Frame Relay frames. Multicasting is part of TCP/IP protocol suite today. Since the unicast IP datagrams can be transmitted over Frame Relay links, there is no reason why multicast IP packets cannot be transmitted. The key to distributing multicast is to enable native multicast support in participating routers. The second part of this chapter documents how IP multicasting and MBone are implemented over the Monterey BayNet. "What the hardware and software requirements are for the Monterey BayNet sites," "how are the routers configured for multicast support" and "where can MBone-related software can be found" are answered. Since uncontrolled multicast packets might saturate the entire network, multicast scope control is an important issue. This chapter also documents how the scope of multicast traffic is controlled over the Monterey BayNet. Other possible solutions are also examined. The chapter concludes with multicast recommendations for Monterey BayNet sites.

B. MULTICASTING WITHIN A FRAME RELAY NETWOTK

Frame Relay has been selected as the WAN connectivity service for the Monterey BayNet because it offers greater access speeds, economy, and a clear transition path for increased bandwidth. (Bigelow, 1995)

1. What is Frame Relay?

Frame Relay is a wide-area network (WAN) protocol which operates at Layer 2 of the Open Systems Interconnections (OSI) model (Figure 5.1). To understand better where Frame Relay fits into the OSI model, it is worthwhile to review the bottom four layers of the OSI model, the primary communication layers. The Physical Layer (Layer 1) is the physical infrastructure, such as wiring and the carrier signal. The Link Layer (Layer 2) is responsible for data transport over a link between machine hardware devices. The Network Layer (Layer 3) is responsible for the routing of data through the network or networks. The Transport Layer (Layer 4) is responsible for end-to-end transport of data.

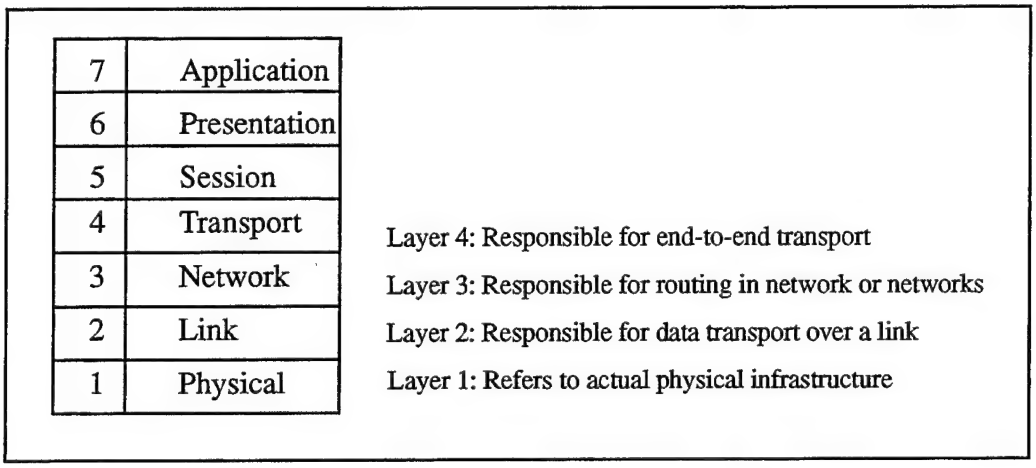


Figure 5.1 OSI Model: Layers 1-4 Associated with Communications

Since Frame Relay is a Layer 2 WAN protocol, it is responsible for the delivery of information over a link (e.g, a WAN link). Figure 5.2 is the demonstration of Frame Relay as a link-layer protocol. It is a packet-based interface standard that has been optimized for the transport of protocol-oriented data. A Frame Relay network consists of user devices and network devices that implement the standard interface. The user device is responsible for delivering

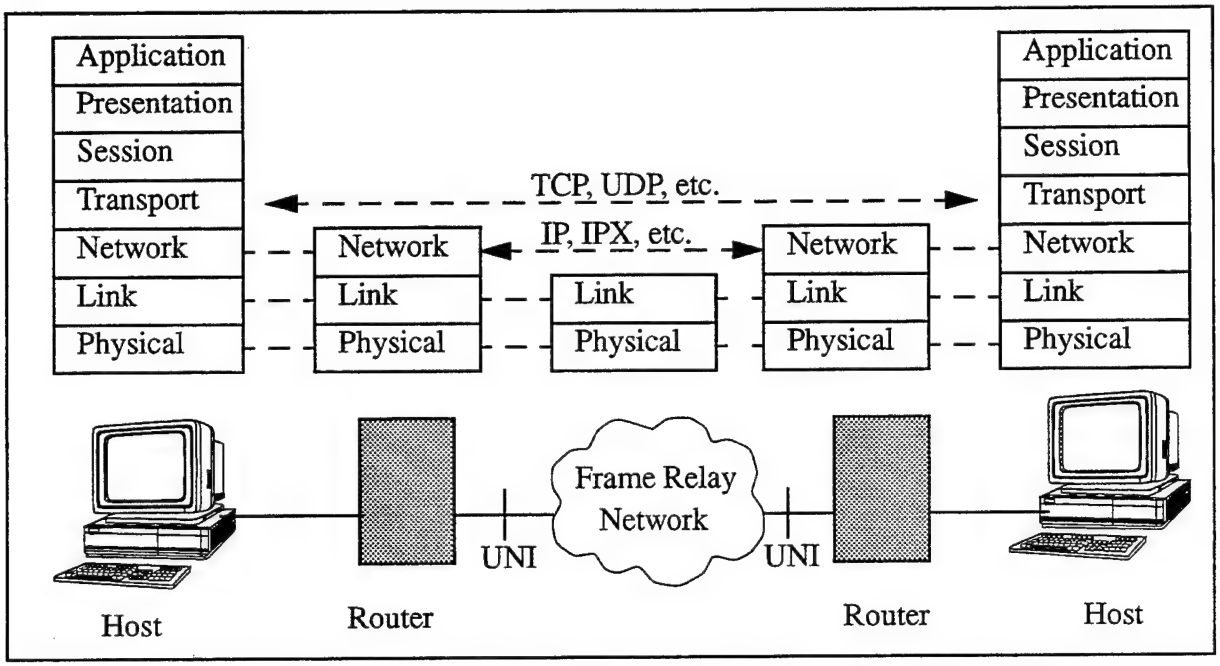


Figure 5.2 Frame Relay as a Link-Layer Protocol after (PacBell, 1994)

frames to the network in the prescribed format. The network device is responsible for switching or routing the frames to the proper destination user device. Since Frame Relay is an interface specification only, the network may route the frames by whatever means the network builders and providers choose. (Frame Relay Forum, 1994) For Monterey BayNet, user devices correspond to routers at schools sites and network devices correspond to off-site PacBell equipment.

Unlike conventional packet switching services (such as X.25) Frame Relay uses statistical multiplexing. The paths, or virtual circuits, are defined through the network. However, bandwidth is not allocated to the paths until actual data needs to be transmitted. The bandwidth is dynamically allocated on a packet-by-packet basis. This reduces internetworking costs. It also provides multiple logical connections within a single physical connection and thereby reduces access costs.

The simplicity of Frame Relay also reduces the computation burden on the network. In Frame Relay networks, error recovery functions are left to the endpoint devices (e.g. PCs and workstations) which run higher-layer protocols. The error-free end-to-end transfer of frames is the responsibility of higher layers. Since out-of-band control signalling is used, meaning that call-control signalling is carried on a separate connection from user data, the multiplexing and switching complexity of the network is reduced. This reduces the amount of network processing needed and helps Frame Relay support the performance and response times of applications.

Frame Relay has well-defined standards approved by ANSI and the ITU/TSS (formerly CCITT). Many equipment vendors and service providers support Frame Relay development, services and standards.

2. How Frame Relay Works

A Frame Relay network is based upon frames with the simple format of Figure 5.3 delivered using an addressing value.(Stalling, 1995) Information of variable length is encapsulated in the frame and sent to the network by user devices. The network uses the addressing information contained in the frame to determine the destination of the frame. The network devices (i.e. Frame Relay switches) read this information and route the frame to the proper destinations. Frames are decapsulated at the destination endpoints.

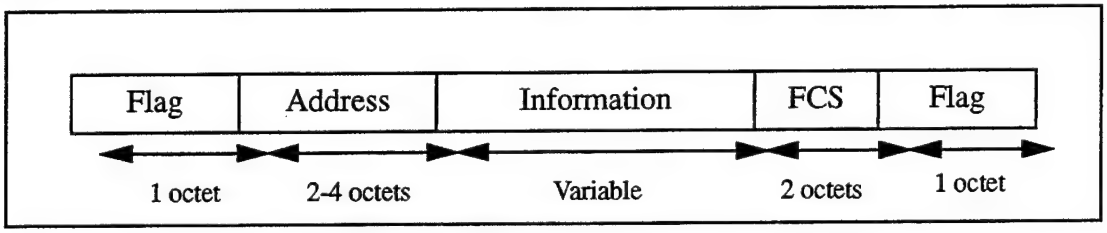


Figure 5.3 Frame Format

What makes Frame Relay so popular is that it is a relatively simple link-layer protocol. It relies on the high-quality transmission links and higher-layer protocols in the endpoint devices. In a Frame Relay WAN, the bad frames are discarded at the link level and the transport layer provides the recovery mechanisms.

As seen in Figure 5.3, there is no control field in a Frame Relay frame format. Only one frame type is available and it is used for carrying data. The frame format does not allow flow control and error control to be performed since there are not any sequence numbers.

The Frame Relay addresses are numbers called Data Link Connection Identifiers (DLCIs). In a Frame Relay network, routing is accomplished by forwarding frames across a permanent virtual circuit (PVC)[1]. Each PVC has an assigned DLCI value. Frame Relay's statistical multiplexing function allows the creation of multiple PVCs on the same physical connection, and

1. Switched Virtual Circuits (SVCs) can also be implemented for Frame Relay networks. However, details of how SVCs are implemented is beyond the scope of this thesis. Since PVCs are implemented for the Monterey BayNet, only PVCs are discussed here.

DLCIs are used to distinguish these multiple PVCs from each other. The DLCI is a logical address and is only understood when combined with the UNI (user-network-interface) on which the PVC is defined. The combination of a UNI and a DLCI on that UNI specifies a PVC endpoint. Two PVC endpoints specify a PVC. The DLCI values at each end of the PVC may be either the same or different. In the former case, DLCI values are managed globally. In the latter case, each end of the logical connection (PVC) assigns its own DLCI from the pool of locally unused numbers. For the Monterey BayNet the former approach is used. All Monterey BayNet DLCI values are listed in (Bigelow, 1995).

The Frame Relay switches maintain and update connection tables. They switch frames from an incoming channel to an outgoing channel based on the appropriate entry in the connection table, translating the DLCI in the frame before transmission. Figure 5.4 illustrates a frame transmission within the Monterey BayNet.

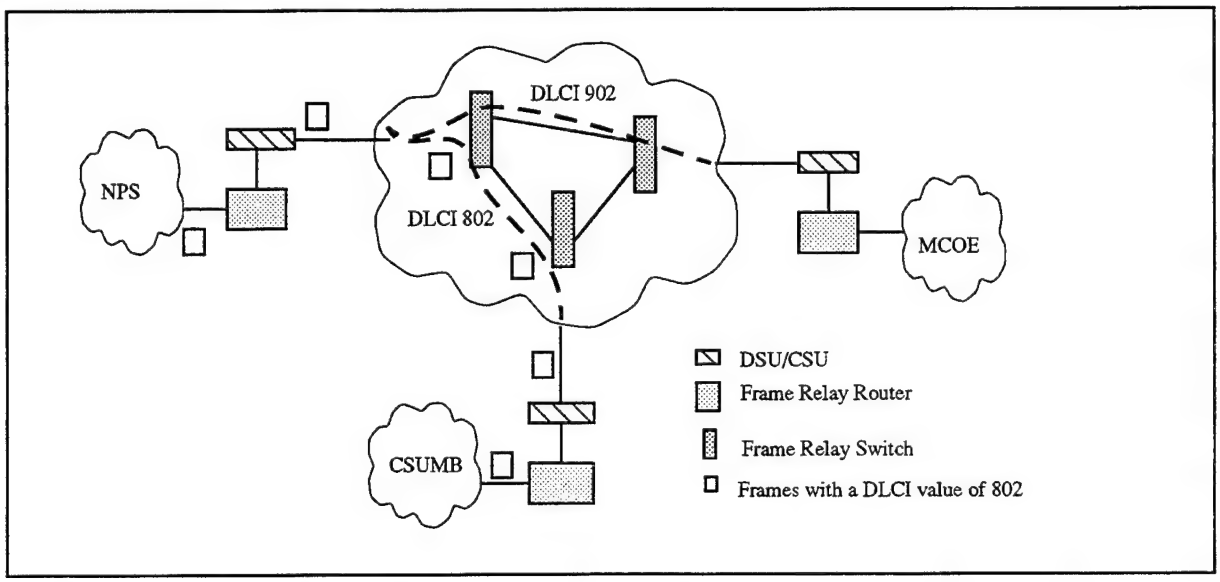


Figure 5.4 Transmission of frames over Permanent Virtual Circuits (PVCs). Frames are forwarded across a PVC to the location specified by the DLCI connection table maintained and updated by Frame Relay switches

Frame Relay networks are capable of carrying network interconnect traffic. Different traffic sources (such as IP and IPX) can be encapsulated into one DLCI. Frame Relay encapsulates its packets within a Q.922 Annex A frame (Brown, 1993). Encapsulating frames contain information necessary to identify the protocol of the carried protocol data unit (PDU). This allows the receiving endpoints to properly process the incoming packet. The Network Level

Protocol ID (NLPID) header of the information field is used to distinguish the different protocols encapsulated within the frame (Figure 5.5).

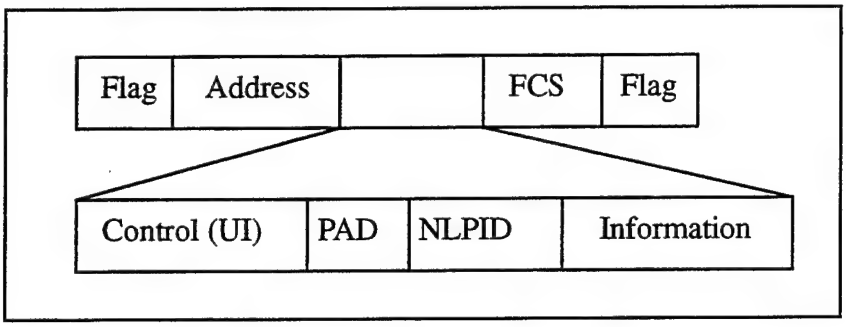


Figure 5.5 Frame Format for Multiprotocol over Frame Relay

The NLPID field is administrated by ISO and ITU/TSS (formerly CCITT) (Brown, 1996). It contains values for many different protocols including IP, IEEE Subnetwork Access Protocol (SNAP) and ISO CLNP (Connectionless Network Protocol). The receiving endpoints determine which encapsulation or which protocol follows by looking at this field.

This feature of Frame Relay is used in Monterey BayNet. The Monterey BayNet was implemented to allow only IP traffic to be routed over the WAN (Bigelow, 1995). The multiprotocol encapsulation feature of Frame Relay is used to send IP datagrams over the Frame Relay network. IP datagrams are encapsulated as described in (Brown, 1996) and for IP over Frame Relay, an NLPID value of 0xCC is used.

Frame Relay networks can be configured and managed by two mechanisms: Local Management Interface (LMI) Rev 1 and ANSI T1.617 Annex D (also known as Annex D) (PacBell, 1994). These two mechanisms are used to monitor the status of the virtual links and determine whether the link is active or inactive. They provide notification of any PVC and DLCI changes (such as addition or deletion). These Frame Relay management protocols also allow the monitoring of the link integrity of the communication link between the network and the user device. DLCI 1023 and DLCI 0 are reserved for the operation of LMI and Annex D. Annex D is an ANSI and CCITT standard and was adopted by the Monterey BayNet (Bigelow, 1995).

3. Multicast Services within Frame Relay Networks

Frame Relay is a connection-oriented protocol. Therefore special implementations are needed for many-to-many multicast services. According to the *Frame Relay PVC Multicast*

Service and Protocol Description (Swallow, 1994), the establishment of Frame Relay multicast service is an administrative operation and requires coordination between the service provider and the service subscriber.

As in other connection-oriented protocols, standardized Frame Relay multicast service relies on a multicast server. The multicast server is a logical entity which provides multicast service to all members. Figure 5.6 is an illustration of a Frame Relay multicast service model. The multicast server is responsible for subscription, unsubscription and the delivery of data units to the members of the active groups. The Frame Relay multicast service model allows either centralized or distributed multicast servers. In this model, the location of the multicast server is unrestricted.

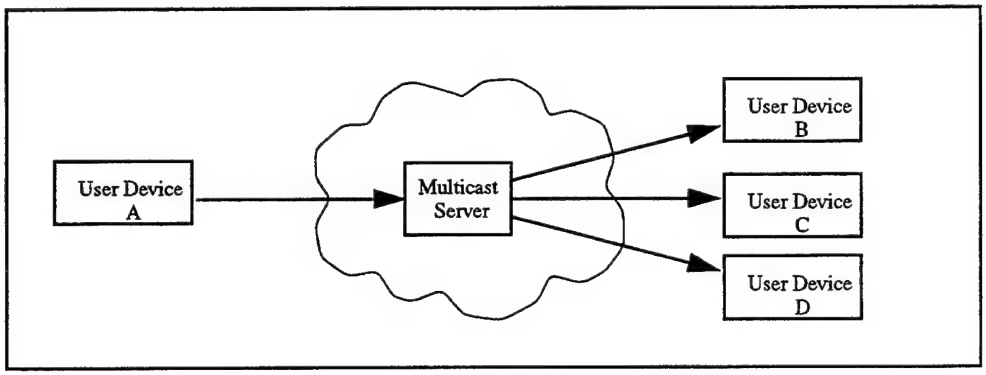


Figure 5.6 Frame Relay Multicast Service Model after (Swallow, 1994)

There are three types of multicast services defined for Frame Relay: one-way, two-way, and N-way multicast services. All require a one-to-many mapping of source to destination but each requires the service provider to interpret the meaning of multiple destinations. (Swallow, 1994)

In a one-way multicast service model, multicast traffic is originated from a root. The root has point-to-point Frame Relay connections (PVCs) to all leaves in the multicast group. The root also has a one-way multicast connection to the multicast server called Multicast Data Link Connection Identifier (Mdlci). Frames are never sent from the network to the root on the Mdlci. Multicast frames are sent to the multicast server via Mdlci by the root. The multicast server accepts frames from the Mdlci and delivers them to each leaf member of the multicast group. As shown in Figure 5.7, frames arrive as though they are delivered on the individual PVCs. Therefore the DLCI in each received frame reflects the source of the message rather than the Mdlci.

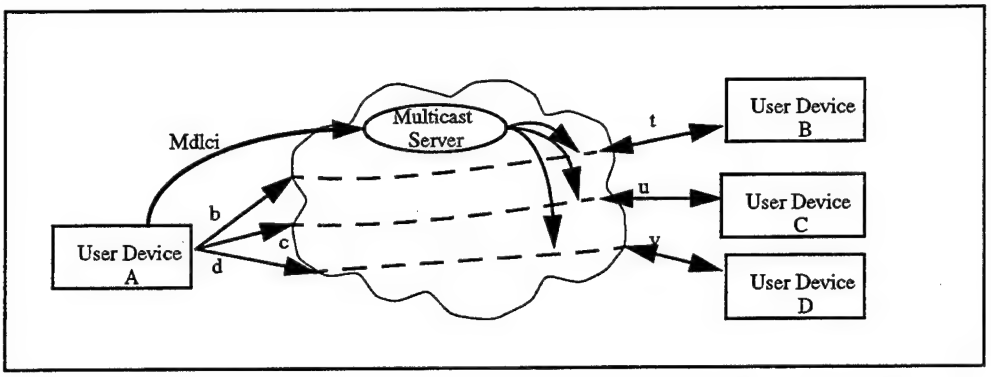


Figure 5.7 One-Way Multicast Service Model after (Swallow, 1994)

This kind of service is useful when the stations are routers and bridges, and also when multicast frames are used for obtaining and verifying the presence or identification of the multicast group members. (Swallow, 1994)

Two-way multicast service provides duplex communication channels. In this model, each participant of the multicast group has a point-to-point Frame Relay connection to the multicast server. One of the participants is defined as the root and the rest of the participants are the leaves. Two-way data transmission is only allowed between the root and the leaves; data is not transferred from one leaf to another. Any data units sent by the root to the multicast server are transmitted only to all leaves in the active multicast group. If a data unit is sent by a leaf, it is only transmitted to the root of the active multicast group, and not to other leaves. Figure 5.8 depicts this service model. This service model is useful in an environment where the point-to-point communication between the root and the leaves is not needed and where the number of leaf stations does not permit the establishment of individual PVCs between the root and the leaves. (Swallow, 1994)

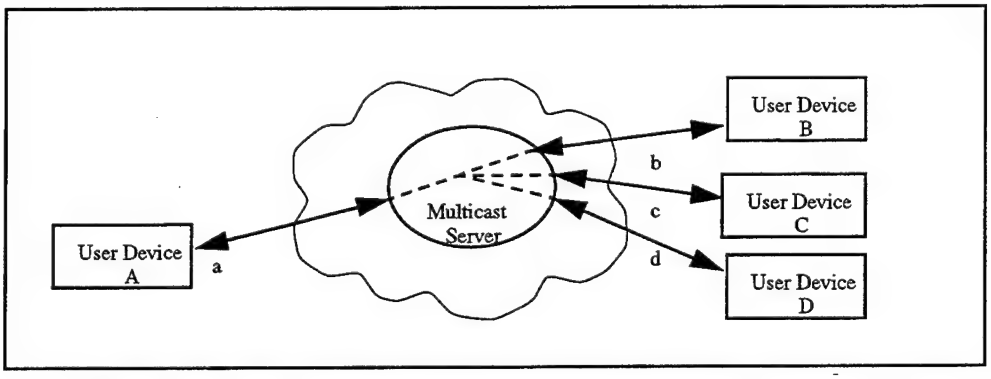


Figure 5.8 Two-Way Multicast Service Model after (Swallow, 1994)

As shown in Figure 5.9, in an N-way multicast service model, all transmissions are duplex and all are multicast. Multicast group members are defined as peers. Any data received by the multicast server is transmitted to all the members of the active multicast group. This kind of implementation of multicasting is suitable for videoconferencing or routing update protocols where all participants require the same data.

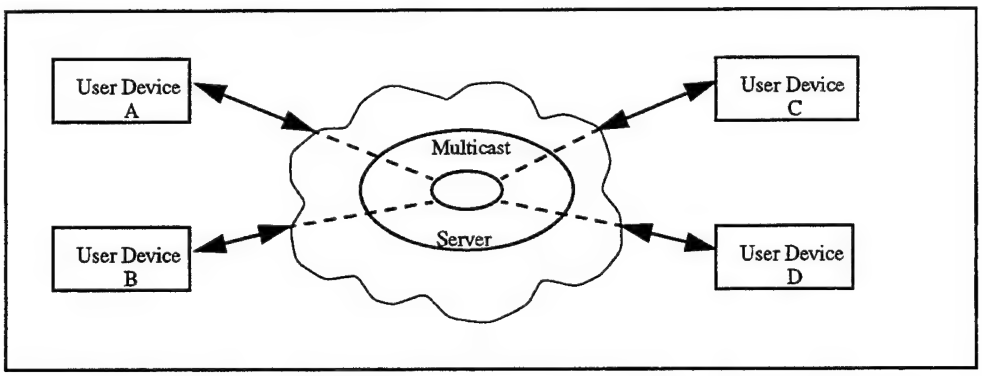


Figure 5.9 N-Way Multicast Service Model after (Swallow, 1994)

To be able to use multicasting for distance learning over Frame Relay networks, an N-way multicast service model needs to be implemented. However, Monterey BayNet (and probably most) Frame Relay networks do not provide any of the multicast service model. Pacific Bell is the service provider of the Monterey BayNet. Since the multicast service is not provided by PacBell, another solution needs to be found. The Cisco routers used by Monterey BayNet sites implement Internet Protocol (IP), Protocol Independent Multicast (PIM) and support native multicast delivery. Therefore, IP multicasting can be deployed for the Monterey BayNet without any need for a multicast server or coordination with PacBell. The next section describes how multicast service was implemented over the Monterey BayNet Frame Relay WAN.

C. MULTICASTING AND MBONE OVER THE MONTEREY BAYNET

As discussed in the previous section, the connection-oriented nature of Frame Relay typically obligates the use of a multicast server for multicast services. However, the Pacific Bell Frame Relay service implementation does not include such a server. Although Pacific Bell acknowledges the need for such servers, there are no plans to provide them for Frame Relay networks. In fact, most Frame Relay service providers do not have this kind of service. A query to the global MBone e-mail list revealed only one Frame Relay network using multicast and that

network did not employ servers either. (Aarnio, 1996)

Since it is not possible to have a multicast server for the Monterey BayNet in the near future and since such an approach might be more costly to the Monterey BayNet sites, a service-provider-independent, already-existing, robust, secure, easy-to-implement and cheap technology needs to be utilized as an alternative.

IP is the only protocol that can be routed within the Monterey BayNet. Unicast IP datagrams are transmitted over PVCs by using the multiprotocol encapsulation feature of the Frame Relay protocol. Multicast is specified as part of the functionality of IP. Multicast Backbone (MBone) provides this functionality. It permits live audio and video transmission across the Internet. Since encapsulated unicast IP packets can also be transmitted over the Frame Relay PVCs, encapsulated multicast IP packets can be transmitted. In order to implement multicasting and MBone over the Monterey BayNet, two sites have been selected as the first test sites: the Naval Postgraduate School (NPS) and the Monterey County Office of Education (MCOE). Multicasting is enabled over the PVCs between these sites after reconfiguring Cisco routers to support PIM, multicasting and MBone. Multicast audio and video was transmitted successfully back and forth between sites.

The following subsections give the general guidelines of how multicasting and MBone can be enabled over the Monterey BayNet as well as the details of how multicasting is enabled between NPS and MCOE.

1. Requirements for the Monterey BayNet Sites

The net design team originally selected the Cisco 2500 family of routers for use by the Monterey BayNet (Bigelow, 1995). Most Monterey BayNet sites have these routers, including NPS which has a Cisco 2503. All routing is handled by the router software. Beginning with the Release 10.2 (2) of the Cisco router software (IOS), multicast support is provided using the Protocol Independent Multicast (PIM) protocol (Cisco, 1996). At the time of this writing, the latest release of the IOS was 11.1(3). IOS releases older than 11.0 do not support multicast, pruning or multicast monitoring tools. Therefore, in order to be able to take full advantage of multicasting, upgrading of router software is essential for the Monterey BayNet sites. Most sites have Cisco IOS Release 10.x. So, upgrading of software is essential. An existing service contract

with Cisco (such as SMARTNet or Comprehensive Maintenance) will provide a password and the ability to freely download the latest version of the Cisco IOS from the Cisco ftp sites. Details for downloading software download and installation are given in (Cisco, 1996). The Cisco 2503 router that NPS has been shipped contains the Release 10.6 of IOS. Since NPS maintains a service contract with Cisco, the version 11.1 (1) of IOS was downloaded and installed in the router. The Monterey County Office of Education (MCOE) bought a new Cisco router and it has been shipped with the latest release of IOS.

In addition to the software upgrade, a hardware upgrade is also needed. Beginning with Cisco IOS Release 10.3, some software image sizes exceed 4 Mbyte (when compressed 2Mbyte). To take advantage of the latest release features, the code or main memory upgrade is needed. To be able to use the IP set provided by the software for Cisco 2500 family routers, 4 Mbyte of code memory and 2 MByte of main memory is required (Cisco, 1996). The software upgrade also requires boot ROM replacement. The current Boot ROM level is 10.2(8a) and it is sufficient for most platforms (Cisco, 1996). The procedure for boot ROM replacement is contained in the Cisco support documentation chapter titled "Replacing the Boot ROMs in Cisco 2500 series and Access Pro PC Card Routers." (Cisco, 1996). Figure 5.10 summarizes the upgrade requirements for the Monterey BayNet sites.

Hardware Requirements	4Mbytes of code memory 2Mbytes of main memory
	Boot ROM Replacement Current level is 10.2(8a)
Software Requirements	Cisco IOS Release 11.1 or greater

Figure 5.10 Upgrade Requirements for Monterey BayNet Sites

2. Router Configuration for Multicast

Cisco IOS has a built-in support for multicast. After the hardware and software upgrade, the routers need to be reconfigured for multicasting because IOS multicasting is disabled by default. It is assumed here that sites already have a connection to Monterey BayNet. If you are not connected to the Frame Relay cloud, please refer first to (Bigelow, 1995) for the basic router configuration. In (Bigelow, 1995), configuration details of the Frame Relay routers (Cisco 2500

family) for unicast IP packet transmission over the Monterey BayNet are given.

The Protocol Independent Multicast (PIM) protocol is the multicast protocol implemented by Cisco routers. Since all sites have Cisco routers, mrouter tunneling mechanisms within the Monterey BayNet are not needed. Similarly, the multicast routing daemon (*mrouted*) installation is not required for any sites. The multicast-capable PIM-enabled routers allow native multicast delivery. In general, as discussed in Chapter IV, not all routers support multicast and so tunneling multicast IP-in-IP is sometimes needed in the larger MBone architecture. For the Monterey BayNet, encapsulation is needed only to deliver IP packets (either unicast or multicast) over Frame Relay PVCs. No tunneling is needed internal to the Frame Relay WAN.

PIM has two modes: sparse and dense modes. Sparse-mode PIM is especially designed for environments where senders and receivers are separated by WAN links. It is suitable for environments in which multipoint data streams go to a relatively small number of LAN segments. For this type of environment, dense-mode PIM uses the bandwidth inefficiently. (Cisco, 1996) Dense-mode PIM is thus not advantageous for Monterey BayNet sites since they have limited bandwidth. Therefore, sparse-mode PIM is selected as the multicast routing protocol for the Monterey BayNet.

Sparse-mode PIM requires a router to be designated as a Rendezvous Point (RP). The rendezvous point collects information about multicast senders and makes that information available to potential receivers. For the sites with the Internet access provided via MCOE, the MCOE Frame Relay router (205.155.43.1) is the RP. For other Monterey County sites, the CSUMB Frame Relay router (137.145.176.1) is the RP. A similar arrangement will need to be provided by UCSC for Santa Cruz county sites.

The primary goal for implementing multicasting over the Monterey BayNet was to have a regional MBone. This requires special administrative control, especially at the NPS site. CSUNet is the Internet Service Provider for the Monterey BayNet. Internet connectivity is provided for Monterey County through CSUMB.(Bigelow, 1995) Addition of an NPS Frame Relay link to the Monterey BayNet raise several important of questions regarding possible routing loops. The 205.155.327.0 subnet is reserved for NPS (Bigelow, 1995). However, there is no such subnet used by NPS. The Frame Relay router is attached to a multicast capable subnet (Figure 5.11). Multicast data is provided to the router from the NPS side via an mrouter, tunnelled to the MBone cloud running on the same subnet. This physical topology puts NPS in a position that it might also

provide Internet connectivity to the Monterey BayNet. As seen in Figure 5.11, NPS is behind a firewall. Frame Relay connectivity causes a back-door for routed traffic into NPS. NPS is also part of MBone cloud. Therefore, enabling multicasting within the Monterey BayNet and making NPS part of it may cause NPS (and also border sites) to receive duplicate multicast packets, and possibly put NPS (and other border sites) in the position of being an MBone provider for the Monterey BayNet. These considerations require the use of a different router configuration for NPS. The router configuration script is provided for NPS as well as for the general Monterey BayNet site configuration in Appendix B.

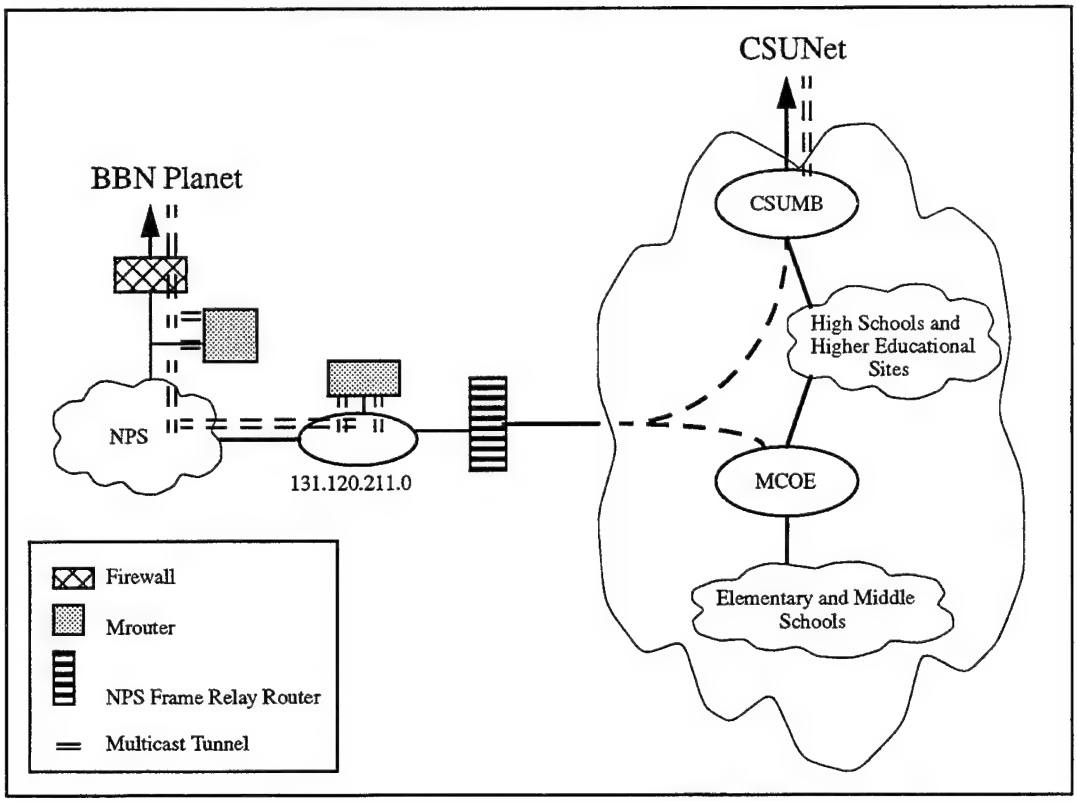


Figure 5.11 Current Topology for NPS and the Monterey BayNet Frame Relay WAN

3. Controlling the Scope of MBone Traffic for the Monterey BayNet

As seen in Figure 5.11, both NPS and CSUMB have MBone connectivity. Enabling multicast within the Monterey BayNet, configuring CSUMB and NPS Frame Relay routers for multicasting and making them part of the regional MBone, might cause two important problems.

First, careless configuration of the router at NPS may cause duplicate multicast packets within the Monterey BayNet. For instance, a global MBone session (such as NASA Space Shuttle

Mission multicast) will be advertised to both CSUMB and NPS Frame Relay routers. If there is a user at La Mesa Elementary School who wants to watch this transmission, this request will be sent to the MCOE Frame Relay router via the La Mesa Elementary School Frame Relay router (see Figure 5.12). The MCOE Frame Relay router will send this request message to upstream routers, which in this case are NPS and CSUMB. Therefore, both NPS and CSUMB Frame Relay routers might attempt to send Space Shuttle Mission multicast packets to MCOE and two copies of the same multicast packet will be received by the La Mesa Elementary School Frame Relay router. In a network where bandwidth is limited, redundant duplicate packet delivery (especially with large data streams like video) is not acceptable. The same problem might occur when a global session is created within the Monterey BayNet. This session will be advertised to MBone cloud via two sites: CSUMB and NPS. This duplicate multicast packet delivery is a serious potential problem for MBone sites that must be properly handled.

Second, a lack of administrative control on the NPS Frame Relay router might put NPS in the unwanted position of providing both Internet and MBone access to the Monterey BayNet. However, CSUMB is the official Internet Service Provider (ISP) for Monterey County sites in Monterey BayNet. BBN Planet (ISP for NPS) does not want to provide Internet access to these sites. Multicast and unicast service to the global Internet needs to come via the same provider. For Monterey County sites in Monterey BayNet, that provider is CSUMB and CSUNet.

Multicast bandwidth always makes ISPs consider whether allow multicast traffic or not. When an experimental multicast tunnel setup between NPS and MCOE was proposed before native multicasting was enabled, the NPS ISP protested, because MBone topology should match normal routing topology. Clearly, controlling multicast traffic passing through the NPS Frame Relay router is an important issue.

To be able to handle these possible problems, the first thing that needs to be done is to determine a topology for MBone connectivity. By keeping in mind the idea of reflecting physical topology for MBone topology, a logical topology is constructed for regional MBone connectivity (Figure 5.12).

The Monterey BayNet is implemented in a way that the Internet access is provided to elementary schools via MCOE, and to sites other than elementary schools (such as high schools, libraries etc.) via CSUMB. High schools and libraries also have Frame Relay connections to MCOE. Instead of allowing multicasting on each PVC, multicasting needs to be allowed only on

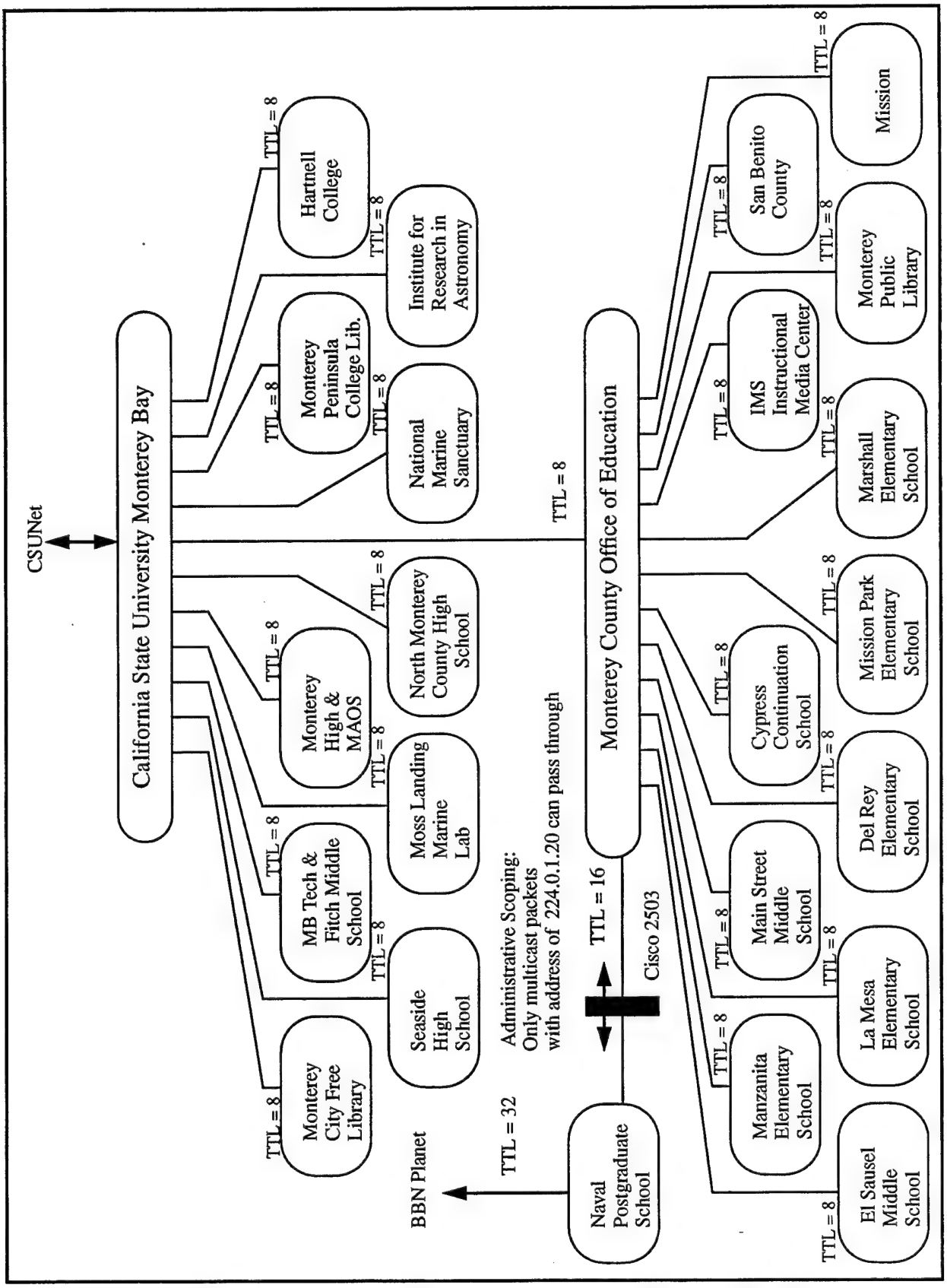


Figure 5.12 Logical MBone Topology for the Monterey BayNet

the PVCs used to access Internet services such as telnet, ftp, www. For instance, Seaside High School has two PVCs: one to CSUMB and one to MCOE. To reflect the IP topology, multicasting should be allowed only on the PVC that is set up to CSUMB. The only exception to this is NPS. NPS has a different ISP. It does not matter on which PVC the multicasting is allowed. Since MCOE was selected as a test site, the multicasting was enabled on the PVC that connects NPS and MCOE.

Preventing NPS from being the Internet Service Provider of the Monterey BayNet was discussed in the previous chapter and the required router configuration is given in Appendix B. NPS might be prevented from being the MBone provider of the Monterey BayNet, and thus avoid duplicate packets, in several ways. The use of administratively controlled multicast group addresses is one way and implemented for the Monterey BayNet.

Frame Relay router's IOS allows administrators to control the multicast traffic which can pass through the router. Access lists are used for this purpose. If a received multicast packet has a group address defined in this access list, it can be accepted and delivered. Otherwise, it is discarded. 224.0.1.20 is an assigned IANA (Internet Assigned Numbers Authority) multicast address reserved for private experiments (Appendix A). This is the only multicast group address for the multicast packets that is permitted to pass through the NPS Frame Relay router. In other words, it is configured in such a way that only multicast packets having multicast group address of 224.0.1.20 can pass through it. For example, assume that a session has been created with a multicast group address of 224.2.139.100. When multicast packets with this group address are received by the NPS Frame Relay router, they are checked against the multicast access list defined in the router. Since they are not allowed to pass through, meaning that, this multicast address is not in the access list, the NPS router discards these packets. No host in the NPS site can receive these packets through the Frame Relay router. Of course, these packets are visible to NPS hosts from the NPS side if they arrived from the upstream mrouter at BBN Planet. With a proper TTL value, multicast packets (with multicast address of other than 224.0.1.20) can pass through the CSUMB Frame Relay router and main campus mrouter and reach NPS hosts via regular MBone tunnels. The same multicast access list is also defined on the Ethernet interface which connects the Frame Relay router to the one of the NPS subnets (131.120.211.0). NPS-originated multicast packets with the multicast address of 224.0.1.20 are the only packets that can pass from the NPS side to the Monterey BayNet side. The multicast packets with the group address other than

224.0.1.20 can reach the Monterey BayNet sites via the NPS main campus mrouter and regular MBone tunnels. Therefore, duplicate packets are almost completely eliminated within the Monterey BayNet.

This scheme poses one deficiency. What happens if a Monterey BayNet user creates a session with the special multicast group address of 224.0.1.20 (which is allowed to pass through the NPS Frame Relay router) and pick a TTL value of 255 which means it can reach the entire world? These packets can be delivered to the MBone cloud by the CSUMB main campus mrouter because of the TTL value. These packets can also pass through the NPS Frame Relay router because they have a legal multicast group address. They also have a TTL value high enough to be forwarded by the NPS main campus mrouter over the tunnel that connects NPS to the MBone cloud. Such a scenario may produce duplicate packets within the global MBone. Therefore, in addition to the use of controlled multicast address, TTL control is also essential. Figure 5.13 summaries the combined use of administratively scoped multicast addresses and corresponding TTL values for scope control. **The one combination not permitted is multicast group address 224.0.1.20 with ttl greater than 32.**

Scope	TTL	Multicast Group Address
Local Site (e.g. single school)	less than 8 (ttl < 8)	224.0.1.20
Regional (Monterey BayNet)	between 16 and 32 (16 < ttl < 32)	224.0.1.20
Global	greater than 32 (ttl > 32)	Any other than 224.0.1.20

Figure 5.13 Controlling the Scope of Multicast Traffic for Monterey BayNet

As seen in Figure 5.13, if the multicast traffic is to be kept local to the subnet, a multicast group address of 224.0.1.20 and a TTL value less than 8 should be selected. Otherwise, routers will be able to deliver these packets to upstream routers. If the multicast traffic is to be kept only within the Monterey BayNet, the TTL value should be between 16 and 32. Such a TTL value is not sufficient for the multicast packets to be delivered to the MBone cloud by either the NPS and CSUMB main campus mrouters. The global transmission of multicast packets require more care because of the considerations discussed previously, such as duplicate packets, etc. A multicast

group address other than 224.0.1.20 and a TTL value of greater than 32 need to be picked. Generally the multicast group address picked automatically by *sd* or *sdr* is sufficient. The CSUMB main campus mrouter is the gateway for the multicast packets with a TTL value greater than 32 and a multicast group address of other than 224.0.1.20.

MBone relies on the TTL values for scope controlling. This control mechanism is completely user dependent. Internet is a free environment. The use of MBone cannot be restricted and also the user cannot be forced to use greater TTL values. Like the conventional scope controlling mechanism, the mechanism proposed here is also user dependent. Nobody can guarantee that the users of the Monterey BayNet will follow the instructions given here. Therefore, administrative control on each Monterey BayNet site and MBone user training remains essential.

4. Alternative Solutions for Scope Controlling

During the design phase of providing multicasting over the Monterey BayNet, the original idea to control multicast scope was to use TTL values. After some discussion and receiving different ideas from MBone mailing list subscribers, the use of administratively controlled multicast group addresses, combined with regular MBone scope controlling mechanism (TTL) was preferred and implemented as discussed in the previous section.

The original idea was little bit different from the regular TTL approach. The mrouters normally decrement the TTL value of multicast packets by 1 before these packets were forwarded over proper interfaces. If a mechanism were available to force mrouters to decrement the TTL value, not by 1 but by an administratively selected higher value for each interface, it might be possible to decrement TTL to different values for incoming and outgoing packets. Then scope controlling might be simpler and user independent. Figure 5.14 shows how administratively decremented TTL values might be used for multicast scope control. For instance, the NPS main multicast router might decrement TTL value of the incoming multicast packets to a value between 0 and 15 so that multicast packets could not pass through the NPS Frame Relay router. Such a functionality would prevent NPS from being the MBone provider of the Monterey BayNet. The multicast packets originating from the Monterey BayNet might pass through the NPS Frame Relay router if they have TTL values between 16 and 31. Since TTL value of these packets were

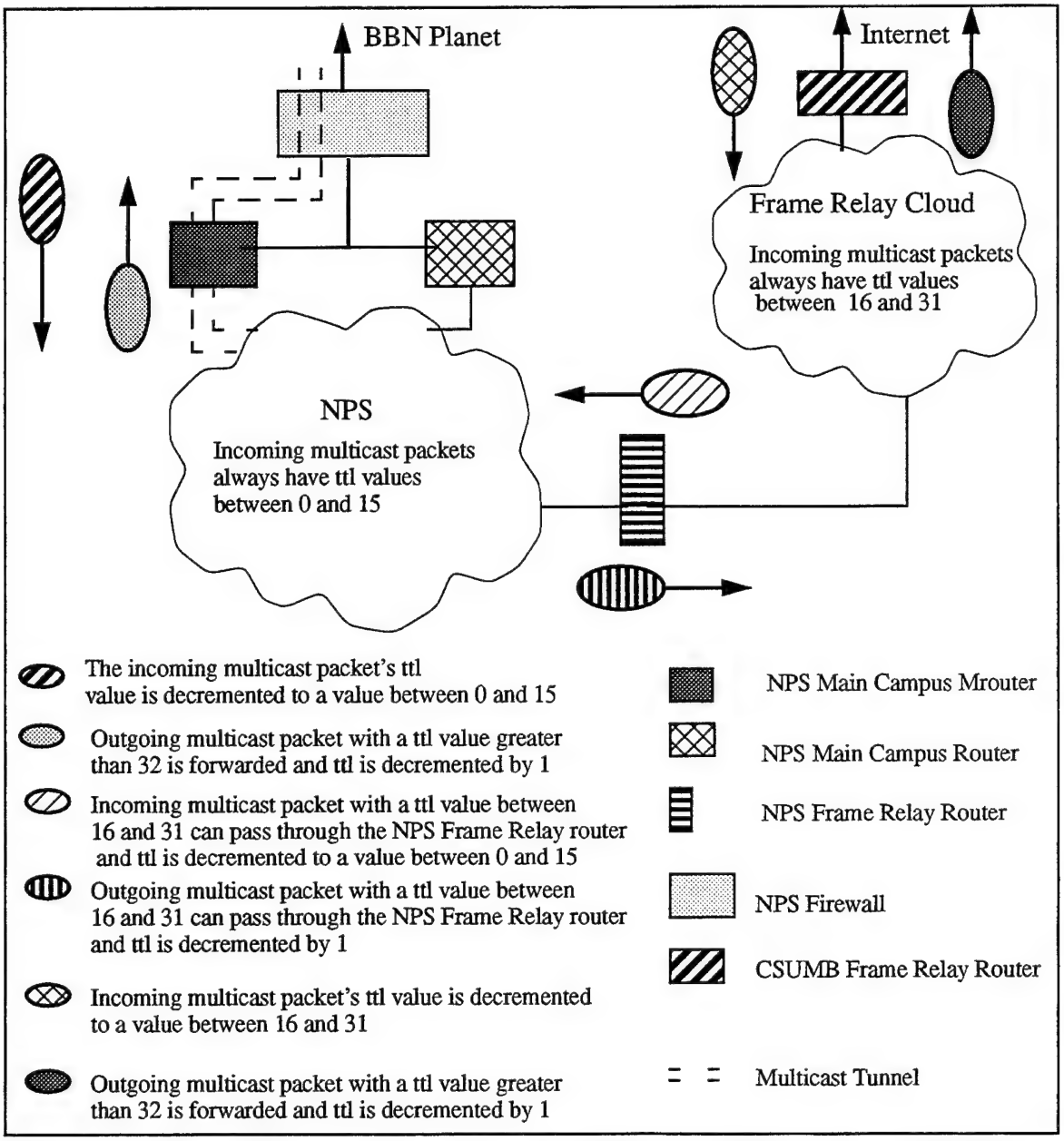


Figure 5.14 Concept Diagram: Desired Use of Administratively Decremented TTL Values for Scope Controlling

decremented to a value between 0 and 15, they would stay only within NPS and could not be forwarded to the MBone cloud because their TTL were not sufficient to be forwarded by the NPS main campus mrouter. Such a scheme remains appealing as a possible solution to prevent users violating the guidelines of Figure 5.13.

Such a solution requires modifications to the existing *mrouted* source code and redistribution of *mrouted* binaries. A modified version of *mrouted* might be used regionally.

However, it does not make sense to do this without any MBone-wide community acceptance and global distribution. Each time a new version of *mrouted* is released, these modifications should be applied to this new release and redistributed regionally. In addition to that, multicast address filtering is again needed. Without any filtering, use of administratively decremented TTL values would not eliminate the possible problems. Therefore, this solution was discarded. It remains an option for future work.

5. Firewall Considerations

Firewalls are used to protect networks against attacks. If the network is behind a firewall, the firewall must be reconfigured for multicast connectivity. Different types of firewalls are used. Each implementation of a firewall differs from the other. Depending of the firewall, the configuration for multicast changes.

The two important protocols used for MBone are IP_PROTOCOL 4 (IP-in-IP) and IP_PROTOCOL 2 (IGMP). As discussed in Chapter IV, if the multicast packets are provided via tunnels, the multicast packets are encapsulated within regular IP packets, such as IP-over-IP. They look like other IP packets to non-multicast-capable portions of the Internet. The firewall should be configured to permit this kind of traffic. Internet Group Management Protocol (IGMP) is the other important protocol for multicast. Multimedia applications for IP use this protocol to join multicast groups. PIM and DVMRP use it to determine the location of hosts that have joined a multicast group. IGMP messages are transmitted within an IP datagram and specified in the IP datagram with a protocol value of 2. (Deering, 1989). Firewalls need to be configured to let the IP packets with the protocol value of 2 pass through. Even though specifying certain protocols (such as IGMP) may not be possible for the firewall, the host running *mrouted* needs to be accessible via unicast to the upstream tunnel provided outside of the local network. All UDP ports on that host need to be opened for any type of authorized access. This is also needed for global monitoring purposes.

The native multicast implementation requires different considerations. As discussed in Chapter IV, Class D IP addresses (ranging from 224.0.0.0 to 239.255.255.255) are used for multicasting. Firewalls need to be configured to allow packets having these addresses in the source field of the IP headers. Correspondingly, IGMP also needs to be allowed to pass through.

Traditional firewalls are used to control only incoming traffic. Outgoing traffic is ordinarily not restricted. However, recent firewall implementations also require similar configurations for the outgoing traffic. Unless the outgoing packets meet the requirements to be able to pass through the firewalls, they are discarded. Therefore, outgoing multicast traffic should be allowed as well as incoming multicast traffic.

The NPS firewall is properly configured for multicast UDP packets, allowing two-way traffic without restriction. However, it is not properly configured for IGMP due to an inadequate firewall software implementation. This restriction prevents IGMP traffic and thus proper multicast monitoring. An important area for future work is to upgrade NPS firewall software and enable IGMP traffic.

6. Installing MBone Tools

MBone tools make live audio and video transmission possible across networks. These tools are available for almost every platform, including PCs. They can be obtained from the Internet without any charge. Unless you have a Macintosh, it is possible to find the right tools suitable for your platform today. Unfortunately, despite repeated queries, it is not clear when the multicast tools will be available for Macintosh platforms.

There are basically four kinds of software tools: announcement tools used to advertise MBone sessions and to launch other MBone tools, video tools used to transmit/receive video, audio tools used to transmit/receive audio, and other tools used to transmit/receive texts and images. All kinds of tools are available for UNIX-based workstations. Since Microsoft delivered the Windows 95 operating system with built-in multicast support in its TCP/IP stack, MBone tools have begun to appear for PCs. Figure 5.15 is the list of tools available for Windows 95 based PCs and Windows NT systems. These tools can be downloaded from the sites listed in Appendix C.

Session directory tools sdr and sd (Figure 5.15) are used to create and announce MBone sessions, as well as to launch other MBone tools. There are no specific system requirements, assuming that you are already connected to a multicast capable network and TCP/IP stack is installed in your PC, to be able to run these tools. vic is the preferred video tool. Additional hardware is not required to be able to receive video. A video capture card and an analog or digital

Announcement Tools	*sd*	Session announcements, launching other tools. *sdr* is currently superseding *sd*
	sdr	
Video Tool	*vic*	Receiving/transmitting video
Audio/video Tool	*nvat*	Combination of audio and video tools
Audio Tools	*vat*	Receiving/ transmitting audio
	rat	

Figure 5.15 MBone Tools Available for Windows'95 Based PCs and Windows NT Systems

video camera are needed for to send video. Currently, *vic* for Windows systems is not capable of sending video; only receiving is possible. *vat* and *rat* are the audio tools. A sound card, speakers, and mike are needed in order to receive/transmit audio. *nvat* is a tool which contains both audio and video components. It can be considered the combination of *vat* and *nv* (another video tool). Its performance on PCs is slow even with 16 MByte of RAM. *sdr*, *sd*, *rat*, *vic* have been tested with the hardware combination listed in Figure 5.16. They work as well as they do in UNIX-based workstations. User manuals are on-line for most tools. These on-line information sources are listed in Appendix C. An early user manual for these tools also can be found in (Emswiler, 1995)

Installation of these tools under Windows 95 is straight forward. They are all in binary data format and distributed in compressed forms (i.e. zip format). Decompression of these files creates an executable file and some dynamic link libraries (*.dll files). Placing these executable files in a directory which is in your path and associated .dll files under the "windows" directory is enough to be able to run these tools. Appendix C also provides additional installation information.

D. RECOMMENDATIONS FOR FUTURE WORK

The bandwidth is limited for the Monterey BayNet since most sites have 128Kbps connections. Limited bandwidth can cause problems for audio and video transmission. In our proposed topology, CSUMB and MCOE are in a position to be the MBone providers for

Operating System	: Windows'95
CPU	: Intel Pentium 90
Main Memory	: 16 Mbyte
Sound Card	: Sound Blaster Pro
Video Card	: Diamond Stealth w/ 1Mbyte of DRAM
Speakers	
Mike	

Figure 5.16 PC Configuration Used for Testing MBone Software Tools

Monterey BayNet sites. The bandwidth available for these sites is 1.536 Mbps. MCOE feeds multicast traffic to 12 sites and CSUMB provides 11 sites. If all sites become part of MBone, this may cause some network congestion problems. For good quality video transmission, 128 Kbps is enough. For example, if 12 sites fed by MCOE are part of the MBone and all want to join the same MBone session. This means that MCOE should use all of its available bandwidth for multicast packet delivery. Oversubscription of the network is highly possible. Experimentation will be needed to see exactly what happens.

Some Monterey BayNet sites still have 56 Kbps line capacity. This is not sufficient capacity for reasonable video and audio quality. It is highly recommended for these sites to upgrade the line capacities to at least 128 Kbps and if possible to 256 Kbps. Without upgrading line capacities, Mbone connectivity will not work. 128 Kbps is likely enough for those sites that already have it. However, if the budget permits, upgrading line capacities to 256 Kbps is worth consideration.

Both the regular MBone scope-controlling mechanism and our proposed scope-controlling scheme for Monterey BayNet regional MBone rely on the users. It is assumed that users are trained and do not use higher TTL values. This is because the current *mrouted* implementation does not provide any other scope control mechanisms (such as address filtering and administratively decremented TTL). This is still an option for future work and worth consideration. Otherwise scope controlling will remain always user dependent.

Since scope controlling is an important issue and user dependent in our proposed

implementation, training for MBone use is essential. MBone is a new technology and still in the experimental phase. Lectures given to system administrators and teachers will improve the efficiency of MBone use and make scope controlling easier.

E. SUMMARY

Frame Relay is a connection-oriented WAN protocol which fits into the Level 2 (Link Layer) of the OSI model. It is a simple protocol that relies on underlying high-quality transmission links and the higher-layer protocols running on the end systems. It offers high access speeds and savings for its users.

A Frame Relay network is based upon the routing of frames by an addressing value. Data is encapsulated in the frame and sent over virtual connections called PVCs. Network devices or Frame Relay switches route the frames to the proper destinations by using the addressing value contained in the frame. Frame Relay addresses called Data Link Connection Identifier (DLCI) are used to distinguish PVCs from each other. DLCI values are assigned either locally or globally. In a Frame Relay WAN.

Frame Relay protocol permits multiprotocol encapsulation. Different traffic sources can be encapsulated into Frame Relay frames. Protocols such as IP and IPX can be routed over Frame Relay networks.

Since Frame Relay is a connection-oriented protocol, multicasting is not easy to support. The establishment of native Frame Relay multicasting is an administrative operation and requires coordination between the service provider and service users. The Frame Relay multicast relies on the multicast servers within the network. Three types of multicast service model are defined for Frame Relay: one-way, two-way, and N-way multicast service models. All require a one-to-many mapping of source to destination but each requires the service provider to interpret the meaning of multiple destinations.

Frame Relay is used as the WAN connectivity service for the Monterey BayNet. Since any of the multicast service models is available for use at Monterey BayNet sites, IP multicasting using the Protocol Independent Multicast (PIM) protocol has been implemented over the network. Implementation of multicasting requires updating the existing technology that Monterey BayNet sites already have. Upgrading the router software, replacing boot ROM on the router, and adding

memory are the basic site needs required in order to implement native IP multicasting over the Monterey BayNet.

Multicast packets can easily saturate the entire network. Therefore, scope control is an important issue. Scoped multicast addresses are used as well as the regular IP multicast scope control mechanism such as the use of TTL) in order to control the scope of the multicast traffic. Such a control is needed in order to avoid duplicate multicast packet delivery both within and outside of the Monterey BayNet.

Implementation of multicasting over the Monterey BayNet requires a different router configuration for the NPS Frame Relay router. It must be configured so that NPS should be prevented from being both the Internet and multicast service provider of the Monterey BayNet.

IP multicasting enables live audio and video transmission across the Internet. Multicast Backbone (MBone) tools make this happen. Those tools are available for Windows platforms today. Implementation of IP multicasting and installation of MBone tools give the K-12 community the ability to remotely join conferences and lectures transmitted over the Internet. These tools were installed into Windows machines at NPS and tested for other Monterey BayNet sites.

The quality of video and audio transmission depends on the available bandwidth. The higher the bandwidth, the higher the quality. Video and audio transmission requires the minimum line capacity of 128 Kbps. For the sites that have 56 Kbps-lines, line upgrade is essential. Successful IP multicast use is not possible without upgrading the line speed to 128 Kbps or better.

VI. MULTICAST WAN MONITORING

A. INTRODUCTION

Even though multicast is an approved part of the TCP/IP protocol suite, the MBone is still a virtual network because many routers do not support multicast. The MBone still relies on numerous UNIX-based workstations on which multicast routing daemon (*mrouted*) is running and tunnels set up between these workstations. For continuous multicast traffic, it is important to monitor the current status of these tunnels and *mrouteds*.

This chapter documents how world wide accessible multicast monitoring pages can be automatically created using public domain monitoring tools, Web capabilities and scripting languages. Public domain multicast monitoring tools (such as *mtrace* and *mrinfo*) are only available for UNIX users. However, most Monterey BayNet sites are schools which have Windows and Macintosh platforms that cannot use these monitoring tools. Combined use of these tools with *perl* and *CGI* scripting make monitoring accessible for Windows or Macintosh users. Automated scripts for world-wide accessible monitoring are provided and evaluated.

B. MULTICAST MONITORING TOOLS: MRINFO AND MTRACE

Monitoring tools provide the ability to determine routes taken by the data packets, to collect statistical information about packet delivery (such as packet losses and rates) and to determine the current status of network components (such as routers and workstations). This kind of information is important for maintaining continuous data communication between hosts and networks.

Multicast routers (mrouters) and tunnels between mrouters are the basic components of IP multicasting. Since most routers are non-multicast capable, additional equipment for multicast packet delivery is still required. As discussed in Chapter IV, mrouters are generally UNIX-based workstations running multicast routing daemon (*mrouted*). Any failure of these workstations means the failure of IP multicasting for the networks fed by those workstations. Multicast packets are delivered from one mrouter to another over virtual links called tunnels. These tunnels are set up over physical Internet connections. For continuous multicast delivery, these tunnels must

always be up.

Multicast monitoring tools are used to monitor this virtual network. *mrinfo* and *mtrace* are the most commonly used tools. They are in the public domain and included in the *mrouted* distributions.

mrinfo is used to gather information about mrouters. It attempts to display the configuration information from the mrouter. If the multicast router responds to the request sent by *mrinfo*, the version number of mrouter and the list of neighboring mrouters are displayed. If the multicast router queried by *mrinfo* has up-to-date software (i.e. Cisco IOS Release 10.3 or greater) the pertinent information (such as metric, threshold and flags) about each interface (physical or logical) is also displayed. This helps to determine the current status of any mrouter and tunnels fed by that mrouter.

mtrace is a tool similar to *traceroute* and is used for tracing unicast paths. *mtrace* is used to determine the path taken by the multicast packets. *mtrace* also accumulates the statistical information along the path. Additional path information, routing errors, packet rates, and packet losses are displayed by *mtrace*. It also collects information related to the tunnel itself (e.g. ttl required for packets to go forward).

C. REQUIRED SYTEM CONFIGURATON FOR USE OF MULTICAST MONITORING TOOLS

Multicast monitoring tools *mtrace* and *mrinfo* are part of the *mrouted* distributions and require super user (root) permissions to be run. This requirement is not caused by security holes in the tools. Rather, raw sockets need to be created by these tools. mrinfo and mtrace use raw sockets to send and receive IGMP messages. To prevent random users from writing their own IP datagrams to the network, only the super user is allowed to create raw sockets (Stevens, 1994). Therefore mrinfo and mtrace need to be run by the super user.Nevertheless, these tools are reliable, useful and secure, so there is no harm done if they are employed by regular users.

Since *mrinfo* and *mtrace* require creation of UDP sockets, they need to be run by the root. By default these tools are owned and run by the root. If they are not run by the root, the execution of the program is terminated. Merely giving execute permission to other users does not make them executable due to their *setuid* bit settings.

Either a set of users or all users can be allowed to run these tools. UNIX file system provides protection over files. Each file has twelve bits, called mode bits, that constitute its mode. In order to allow regular users to run these tools as well as the root, the mode bits must be set properly. Nine of the mode bits are used to control who can read, write, and execute the contents of the file. Figure 6.1 shows the read, write, and execute permissions of *mrinfo* and *mtrace*. According to the permission bit settings, these tools can supposedly be run by anybody. However, *setuid* settings of *mrinfo* and *mtrace* restrict access only to the root.

```
-rwxr-xr-x 1 root root 123183 Jul 30 18:45 mtrace*
-rwxr-xr-x 1 root root  91165 Jul 30 18:48 mrinfo*
```

Figure 6.1 Permission Bits Settings of *mtrace* and *mrinfo*

In order to permit any user other than root to run these tools, the setuid bits must be set. Three mode bits other than the nine permission bits are used to control program execution. The setuid bit is one of them and allows programs to access files and processes that would be otherwise off limits to the user that runs them (Nemeth, 1995). The details of how the setuid bit is set are given in Appendix D. After the setuid bit is reset, anybody can run *mtrace* and *mrinfo*.

To be able to be run by anybody is an important issue for world-wide accessible multicast monitoring tools. *CGI* scripts allow programs to be run via a Web browser, such as Netscape. When these scripts are run via a browser, they are run with a permission of *nobody* (owner of nothing) permission. *nobody* is the owner of software which does not require any special permissions (Nemeth, 1995). The world-wide accessible monitoring tools that will be introduced in the following section are written in the *perl* scripting language (Wall, 1991). They simply run *mrinfo* and *mtrace* and can be invoked via a browser. Therefore *mtrace* and *mrinfo* need to be runnable by anybody (including *nobody*). Setting *setuid* bit is the only requirement for world-wide accessible multicast monitoring tools. Such setting is safe and needed for use of multicast monitoring tools. Figure 6.2 shows the mode bit settings of *mtrace* and *mrinfo* that can be run by anybody.

```
-rwsr-xr-x 1 root root 123183 Jul 30 18:45 mtrace*
-rwsr-xr-x 1 root root  91165 Jul 30 18:48 mrinfo*
```

Figure 6.2 Mode Bit Settings of *mtrace* and *mrinfo* That Can Be Run by Anybody

As an alternative to giving anybody permission to run *mtrace* and *mrinfo*, only a set of users can be allowed to run these tools. Such a configuration is not enough for world-wide accessible multicast monitoring tools. However, for experimental use this type of permission is needed. For instance, using the Automated Mrouter Checking Program (AMCP) introduced in the next section, such a configuration is sufficient. If only a set of users are allowed to use these tools, a group must be created for those users first.

To support group access UNIX provides the ownership concept. The owner of a file has control over it. Files can be owned by either a single user or a set of users called a group. After the setuid bit is set and a group is created, the group ownership of *mtrace* and *mrinfo* should be changed. The permission bits must be set so that only the root and that group can run these tools. At NPS, for experimental purposes, a group named studRoot has been created and students that belong to this group have been allowed to run these tools. Appendix D details how this group was created, how group ownership of *mrinfo* and *mtrace* has been changed for that group, and how permission bits have been set. Figure 6.3 shows the mode bit settings after the studRoot group was created and the group ownership of *mrinfo* and *mtrace* was changed.

```
-rwsr-xr-- 1 root studRoot 123183 Jul 30 18:45  mtrace*
-rwsr-xr-- 1 root studRoot  91165  Jul 30 18:48  mrinfo*
```

Figure 6.3 Mode Bit Settings of *mtrace* and *mrinfo* That Can Be Run by Root and Group studRoot

Monitoring tools such as *mtrace* and *mrinfo* use IGMP messages to gather the desired information. As discussed in Chapter V Section C.6 Firewall Considerations, if a site is protected by a firewall, the firewall must be configured to permit IGMP traffic. Unless the firewall is configured for IGMP traffic, monitoring tools cannot gather information about the multicast traffic outside of the firewall. Therefore, if you intend to use these tools for monitoring multicast sources outside of the network, be sure the firewall is configured properly for multicasting and IGMP traffic. Such reconfiguration remains to be done at NPS when firewall software is upgraded.

D. AUTOMATION AND WORLD-WIDE ACCESSIBILTY

1. Automated Mrouter Checking Program (AMCP)

Continuity of multicast traffic during a conference transmission over the MBone is an important issue. To lose multicast connectivity in the middle of the transmission is very bad for both the source and the receivers of the transmission. At the same time, assessing problems in the distribution of multicast traffic can be extremely difficult. Finding the point of failure and what is wrong with the network may require direct control over the network. Such control is impossible when working globally. To have someone monitoring the network all the time is also an unfeasible solution. The right answer is an automated monitoring mechanism.

The Automated Mrouter Checking Program (AMCP) is a script written in *perl*. It simply runs *ping* and *mrinfo* monitoring tools and records their results as .html pages. *ping* is a tool used to determine if a host is dead (unreachable) or alive. In order to determine the current status of the host running *mrouted* (the mrouter), AMCP runs *ping*. If the program gets a response from the mrouter it concludes that the mrouter is alive, otherwise the mrouter is dead (or unreachable) and therefore *mrouted* is not running. If the mrouter is alive, AMCP then runs *mrinfo* to determine if the *mrouted* is running on that mrouter, since an mrouter can only respond to an *mrinfo* request if *mrouted* is running. Only if AMCP gets an *mrinfo* response from the mrouter does it conclude that *mrouted* is running. AMCP reads information about mrouters from a file called `mrouter.info`. A sample `mrouter.info` file used for NPS local MBone is in Figure 6.4.

#Host_Name	Host_IP_Adress	Physical_Location	Point_of_Contact
#			
mbone.nps.navy.mil	131.120.254.59	Computer_Center	romo@nps.navy.mil
mbone.cc.nps.navy.mil	131.120.53.21	Computer_Center	romo@nps.navy.mil
cadet.stl.nps.navy.mil	131.120.64.17	STL_Lab.	mcgredo@stl.nps.navy.mil
intruder.aa.nps.navy.mil	131.120.149.55	H-103	tony@nps.navy.mil
ntc.nps.navy.mil	131.120.57.3	Computer_Center	ingram@nps.navy.mil
131.120.141.100	131.120.141.100	Auditorium	blau@nps.navy.mil
noise.usw.nps.navy.mil	131.120.140.62	R-107	hudson@usw.nps.navy.mil
indigo1.me.nps.navy.mil	131.120.151.221	ME_Comp_Lab	marco@me.nps.navy.mil
chandra.ece.nps.navy.mil	131.120.20.39	SP-308	voigt@ece.nps.navy.mil
auvonyx.me.nps.navy.mil	131.120.7.112	Golf_Course	marco@lex.me.nps.navy.mil
betelgeuse.cs.nps.navy.mil	131.120.211.3	SP-500	whalen@cs.nps.navy.mil
zeta.nps.navy.mil	131.120.254.222	Computer_Center	romo@nps.navy.mil
utumno.barrnet.net	131.119.244.11	Stanford_University	jhawk@bbnplanet.com

Figure 6.4 Sample `mrouter.info` File

The AMCP reads information from the `mrouter.info` file and sends *ping* and *mrinfo* messages for each mrouter that has an entry in the file. If it detects the mrouter is dead or *mrouted* is not running, it logs this information. The first time it detects that the mrouter is dead, or *mrouted* is not running, it records the name of the mrouter and the time into a file named <mroutername>.down and sends an e-mail to the corresponding person specified as the point of contact in the mrouter.info file. Figure 6.5 is the mail sent to the system administrator by AMCP run for NPS. This mail is sent only the first time the mrouter is detected as dead.

```
This is a report generated by Automated Mrouter Checking
Program (AMCP)!

The mrouted (multicast routing daemon) is normally running
on host cadet@stl.nps.navy.mil (131.120.64.17) for your
subnetwork.

The AMCP program detected on Tue Sep 9 09:00:05 PDT 1996
that cadet@stl.nps.navy.mil is not alive and mrouted is
not running.

Please check it and make mrouted running to restore
Multicast Backbone (MBone) connectivity.

The current status of all mrouters on the NPS Campus can
be found at
http://www.stl.nps.navy.mil/~erdogan/mbone/report.html

This page is updated at one hour intervals. You will not
receive further reports unless cadet@stl.nps.navy.mil
status changes.

Further unicast connectivity status is available at
http://www.stl.nps.navy.mil/~iirg/atm/monitoring/
       Ping_pages/NPS/status.html

Thank you.
P.S. For more information about these projects, please see
the Information Infrastructure Research Group at
http://www.stl.nps.navy.mil/~iirg
or contact Don Brutzman brutzman@nps.navy.mil
```

Figure 6.5 Sample Electronic Mail Sent to Point of Contacts by AMCP

If AMCP is run after the mrouter is detected as dead for the first time and the mrouter is still dead, it just appends the status information into the log file. If a status change is detected (such as the mrouter is recovered) the log file is renamed as <mroutername>.down.log for later inspection. The same steps are followed depending on the *mrinfo* responses received from the mrouter. AMCP always generates a Web page reflecting the current status of both the mrouter and the *mrouted*. In addition to monitoring results, the information such as host name, host IP address, physical location, and point of contact (given in the `mrouter.info` file) are displayed. Figure 6.6 is the Web page created by AMCP running for the local NPS MBone.

To monitor the local NPS MBone, the AMCP *perl* script is run by *cron* on an hourly basis. *cron* is a UNIX daemon which permits automation of program execution. Programs can be run by *cron* at user-specified times. AMCP is automatically run by *cron* on an hourly basis. The status of all mrouters around NPS is monitored and status information is logged on an hourly basis. This provides complete current status of the local NPS MBone.

Automation of such a program allows the detection of a problem caused by a dead mrouter or killed *mrouted* shortly after it occurs. The shorter the time interval that AMCP is run by *cron*, the higher the probability of detecting the problem as soon as it occurs. ACMP has been run for the local NPS MBone very successfully. If this program is simultaneously run at different sites for the Monterey BayNet, full diagnostic monitoring of multicast traffic over the Monterey BayNet is possible. When it is run by multiple sites, the failure of one site does not prevent effective multicast monitoring. *perl* is available for most platforms, even for Windows and Macintosh. However, automation provided by *cron* is UNIX-specific. Therefore MCOE and CSUMB are eligible sites since both have UNIX workstations. The AMCP *perl* source code is provided in Appendix E.

2. mrinfo and mtrace Gateways

mrinfo and *mtrace* are written for UNIX-based platforms. At the time of this writing, not all MBone-related software (but some MBone tools) are imported to Windows and Macintosh platforms. After Monterey BayNet sites running Windows and Macintosh platforms join the MBone, they will need to monitor multicast traffic. Since tools have not yet been available for these platforms, they apparently cannot monitor the multicast traffic. However, *perl*-based *CGI*

URL of this page is *http://www.stl.nps.navy.mil/~erdogan/mbone/report.html*

This report was generated by AMCP on Thu Sep 12 09:05:30 PDT 1996

Mrouter Status Report for NPS

Mrouter Host Name	Mrouter IP Adress	Mrouter Physical Location	Status of Mrouter	Status of Mrouted	Point of Contact
mbone.nps.navy.mil	131.120.254.59	Computer_Center	alive	running	romo@nps.navy.mil
mbone.cc.nps.navy.mil	131.120.53.21	Computer_Center	alive	running	romo@nps.navy.mil
cadet.stl.nps.navy.mil	131.120.64.17	STL_Lab.	alive	running	mcgredo@stl.nps.navy.mil
intruder.aa.nps.navy.mil	131.120.149.55	H-103	alive	running	tony@nps.navy.mil
ntc.nps.navy.mil	131.120.57.3	Computer_Center	alive	not responding, possibly not running	ingram@nps.navy.mil
131.120.141.100	131.120.141.100	Auditorium	alive	running	blau@nps.navy.mil
noise.usw.nps.navy.mil	131.120.140.62	R-107	alive	not responding, possibly not running	hudson@usw.nps.navy.mil
indigo1.me.nps.navy.mil	131.120.151.221	ME_Comp_Lab	alive	running	marco@me.nps.navy.mil
chandra.ece.nps.navy.mil	131.120.20.39	SP-308	alive	not responding, possibly not running	voigt@ece.nps.navy.mil
auvonyx.me.nps.navy.mil	131.120.7.112	Golf_Course	not alive	not running	marco@lex.me.nps.navy.mil
betelgeuse.cs.nps.navy.mil	131.120.211.3	SP-500	alive	running	whalen@cs.nps.navy.mil
zeta.nps.navy.mil	131.120.254.222	Computer_Center	alive	running	romo@nps.navy.mil
utumno.barrnet.net	131.119.244.11	Stanford_University	alive	not responding, possibly not running	jhawk@bbnplanet.com

Point of contact: *erdogan@cs.nps.navy.mil*

Figure 6.6 Report Generated by AMCP on an Hourly Basis

scripts allow users to remotely run programs via Web page interfaces. *mrinfo* and *mtrace* can be run on remote UNIX servers so that even schools can monitor multicast, reading the results via Web pages.

mrinfo and *mtrace* gateways are provided here to meet such a need. These are *CGI/perl* scripts. They simply run *mrinfo* and *mtrace* to gather information and then display the results to users on via Web page. Like many UNIX programs, *mrinfo* and *mtrace* can also be run from a command line. *mrinfo* and *mtrace* Gateways provide a Graphical User Interface (GUI) to the users of these tools and make their use easier. Figure 6.7 and Figure 6.8 are the GUIs created by these programs.

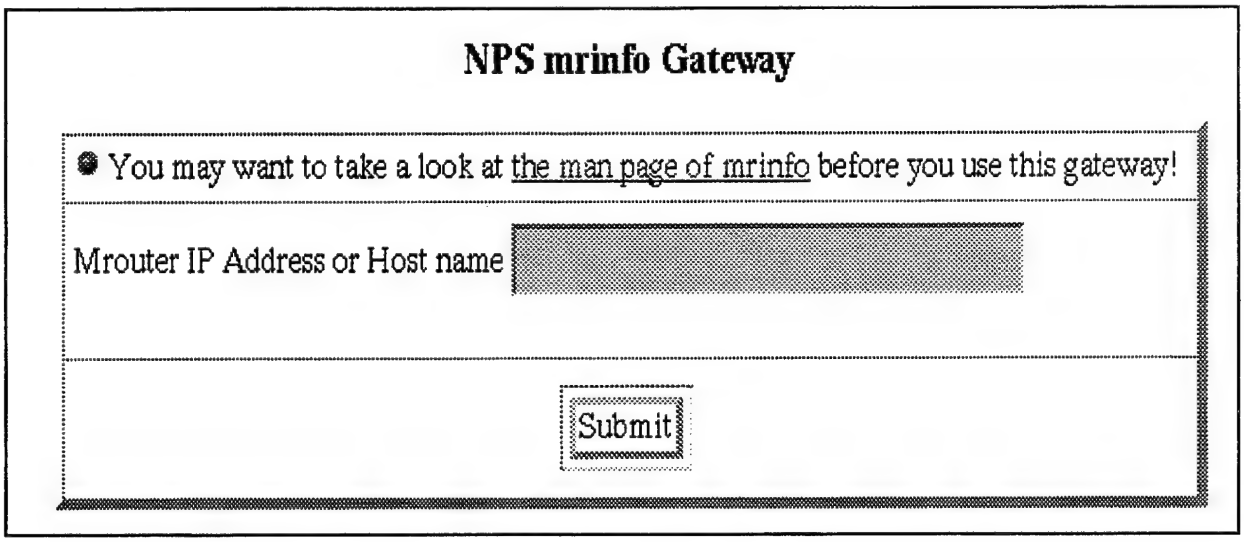


Figure 6.7 HTML GUI for *mrinfo* Gateway

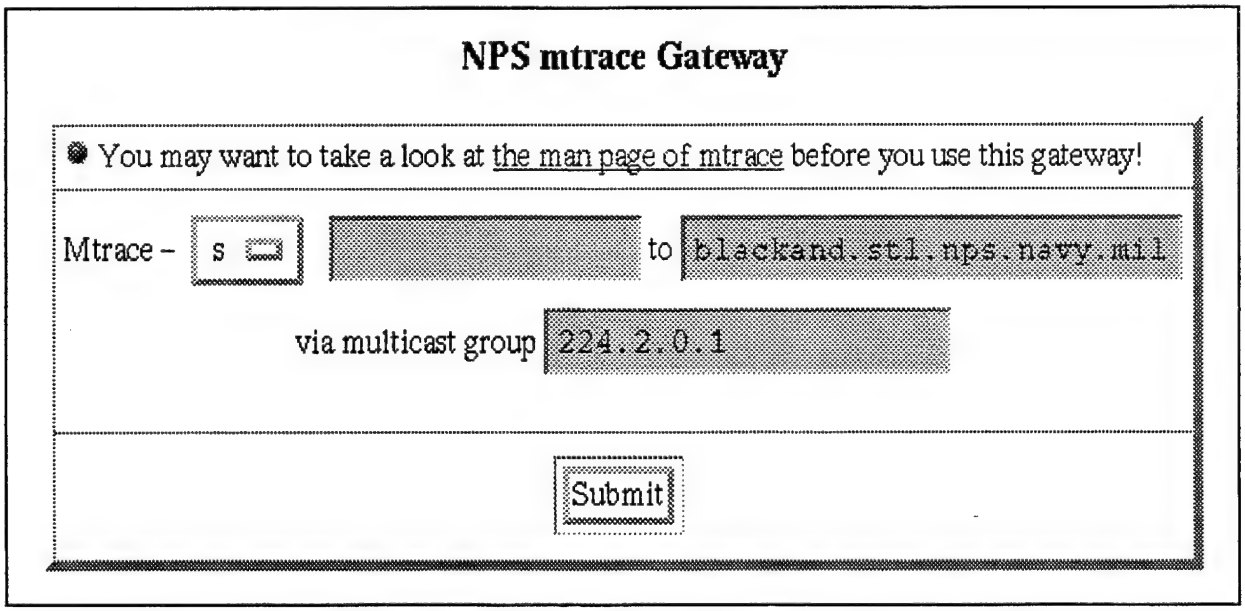


Figure 6.8 HTML GUI for *mtrace* Gateway

The *mrinfo* gateway page displays the configuration details from an mrouter like *mrinfo* itself does. The *mrinfo* gateway is able to add hyperlink functionality provided by regular Web pages. It creates hot links to each mrouter to which the queried mrouter is tunneled. By clicking on these hot links, users can easily jump from one mrouter to another and see the configuration details of each mrouter. Figure 6.9 is a sample output for a single *mrinfo* Gateway query. The provided configuration is for the NPS main campus mrouter (*mbone.nps.navy.mil*). The UNIX manual page for *mrinfo* is also provided for the users as part of the program. Users can learn how *mrinfo* works and what it provides. The *mrinfo* man page is included in Appendix F.

This report was generated on Thu Sep 5 10:59:22 PDT 1996

Configuration Details for mbone.nps.navy.mil

131.120.254.59 (mbone.nps.navy.mil) [version 3.8,prune,genid,mtrace]:
131.120.53.21 -> 0.0.0.0 (local) [1/1/querier/leaf]
131.120.254.59 -> 131.120.254.222 (zeta.nps.navy.mil) [1/1]
131.120.254.59 -> 131.120.254.57 (star.nps.navy.mil) [1/1]
131.120.254.59 -> 131.119.244.11 (utumno.barrnet.net) [1/32/tunnel]
131.120.254.59 -> 192.31.48.211 (192.31.48.211) [1/32/tunnel/down/leaf]
131.120.254.59 -> 134.89.64.1 (algae.mbari.org) [1/8/tunnel/leaf]
131.120.254.59 -> 131.120.142.126 (pine.or.nps.navy.mil) [1/1/tunnel/down/leaf]
131.120.254.59 -> 131.120.149.55 (intruder.aa.nps.navy.mil) [1/1/tunnel]
131.120.254.59 -> 131.120.57.3 (ntc.nps.navy.mil) [1/1/tunnel/down/leaf]
131.120.254.59 -> 131.120.141.100 (131.120.141.100) [1/1/tunnel]
131.120.254.59 -> 131.120.140.62 (noise.usw.nps.navy.mil) [1/1/tunnel/down/leaf]
131.120.254.59 -> 131.120.151.221 (indigo1.me.nps.navy.mil) [1/1/tunnel]
131.120.254.59 -> 131.120.20.39 (chandra.ece.nps.navy.mil) [1/1/tunnel/down/leaf]
131.120.254.59 -> 131.120.150.143 (auvonyx.me.nps.navy.mil) [1/1/tunnel/down/leaf]
131.120.254.59 -> 131.120.211.3 (131.120.211.3) [1/1/tunnel/leaf]
131.120.254.59 -> 198.189.249.186 (MBONEIndy.monterey.edu) [6/16/tunnel/down/leaf]

Back to Mrinfo Gateway
URL : *http://www.stl.nps.navy.mil/~erdogan/mbone/mrinfo_gw.cgi*
Point of contact : *erdogan@cs.nps.navy.mil*

Figure 6.9 Report generated by *mrinfo* gateway. Each link leads to similar *mrinfo* reports for linked mrouters

The *mtrace* gateway is also a *perl/CGI* script. It simply displays the route taken by multicast packets from a source to a destination. The only required argument for *mtrace* gateway is the destination information either the hostname or the IP address of the destination host. The default source is the machine on which *mtrace* gateway is run. The *mtrace* gateway, with its source code provided in Appendix E, is run on one of the NPS hosts (`www.stl.nps.navy.mil`). In order to run *mtrace* gateway on a different machine, you must also change the default source machine for which the *mtrace* gateway will be run. Figure 6.10 is a sample output of the *mtrace* gateway. It shows the multicast path from `blackand.stl.nps.navy.mil` to the NPS main campus mrouter `mbone.nps.navy.mil` and the statistical information throughout the path. The man page for *mtrace* is also provided as part of the program and in Appendix G.

```
        This report was generated on Thu Sep 5 11:16:22 PDT 1996
```

Multicast route from mbone.nps.navy.mil back to blackand.stl.nps.navy.mil

```
Mtrace from 131.120.254.59 to 131.120.63.25 via group 224.2.0.1
Querying full reverse path...
 0  blackand.stl.nps.navy.mil (131.120.63.25)
-1  gate-cadet.stl.nps.navy.mil (131.120.63.17)  DVMRP  thresh^ 1
-2  ? (131.120.211.3)  DVMRP  thresh^ 1
-3  mbone.nps.navy.mil (131.120.254.59)  DVMRP  thresh^ 1
-4  mbone.nps.navy.mil (131.120.254.59)
Round trip time 35 ms
Waiting to accumulate statistics... Results after 10 seconds:
   Source         Response Dest    Packet Statistics For  Only For Traffic
131.120.254.59  224.0.1.32       All Multicast Traffic  From 131.120.254.59
     v       __/  rtt   29 ms    Lost/Sent = Pct  Rate    To 224.2.0.1
131.120.254.59  mbone.nps.navy.mil
     v     ^      ttl    1         0/755  =  0%  75 pps  0/0    = --%    0 pps
131.120.211.3   ?
     v     ^      ttl    2       692/692  =100%  69 pps  0/0    = --%    0 pps
131.120.64.17
131.120.63.17   gate-cadet.stl.nps.navy.mil
     v      \__   ttl    3          0            0 pps    0             0 pps
131.120.63.25   131.120.63.25
  Receiver      Query Source
```

Back to Mtrace Gateway

URL : *http://www.stl.nps.navy.mil/~erdogan/mbone/mtrace_gw.cgi*

Point of contact : *erdogan@cs.nps.navy.mil*

Figure 6.10 Report Generated by *mtrace* Gateway

E. SUMMARY

Implementing multicasting over the Monterey BayNet requires monitoring of multicast connectivity. Continuity in multicast data transmission is a requirement for live audio and video transmission. Losing multicast connectivity right in the middle of a conference transmission is not acceptable for either the sender or the receiver. Monitoring tools are needed to figure out when things go wrong with the network and to diagnose the problems. Commercial tools are expensive and the public domain tools are cryptic.

The multicast monitoring software tools described in this chapter were developed to provide an economical solution to the multicast monitoring problem. The Automated Mrouter Checking Program (AMCP) automates the monitoring process. It is run on an hourly basis by a UNIX daemon called cron. It monitors the status of multicast routers around NPS and generates a report in HTML format which is accessible world-wide. It also has the ability to alert the system administrators about the problems detected by sending electronic-mail. It successfully monitors the local NPS MBone.

Public domain multicast monitoring tools are written for UNIX-based platforms. They are not available for use by K-12 community that have Windows and Macintosh platforms. *mrinfo* and *mtrace* Gateways were developed to meet monitoring needs of Monterey BayNet sites. These tools are written in *CGI/perl* and can be run via Web pages. *mrinfo* Gateway is used to query any mrouter and get the configuration information from the mrouter. *mtrace* Gateway is used to gather information about the path taken by the multicast packets. These tools are accessible world-wide and can be used by all Monterey BayNet sites.

VII. EXPERIMENTAL RESULTS

A. INTRODUCTION

This chapter provides the experimental results. Originally, it was planned to perform some experiments during the Web Content and Access Workshop Monterey 1996 that was held at Naval Postgraduate School on August 23/26, 1996. However, since some administrative problems were not resolved in time, this workshop was not multicast over Frame Relay. For test purposes, TTL-controlled MBone sessions were created and the performance of MBone over Frame Relay connections was evaluated later. The results shows that MBone is possible even on low-speed (128 Kbps) Frame Relay connections and low-cost personal computers.

B. PHASE I: TESTING MULTICAST ON THE FRAME RELAY PVC WHICH IS SET UP BETWEEN NPS AND MCOE

Not all Monterey BayNet sites are ready for the implementation of MBone. Most Monterey BayNet sites have Macintosh platforms. Unfortunately MBone software was not available for Macintosh platforms at the time of this writing. In addition to that, as discussed in Chapter V, hardware and software upgrades are essential for the already-installed Frame-Relay-capable Cisco routers. Therefore multicasting was enabled only between two test sites: Naval Postgraduate School (NPS) and Monterey County Office of Education (MCOE). Because of the limited budget available to Monterey BayNet sites, the implementation of multicasting over the Monterey BayNet may take longer than is expected.

In order to show that multicasting and MBone can be implemented over Monterey BayNet, two Monterey BayNet sites, NPS and MCOE, were selected as test sites. Multicast, especially Mbone, was not a new issue for the NPS site. People at NPS were familiar with MBone and knew what could be done with the MBone. However, MBone was a totally new technology for MCOE. When people saw what the MBone could provide, they were excited. They have pointed out that it would be great to be able to use the MBone for educational purposes. Since people at these sites were excited about the MBone, most administrative problems were solved easily.

Multicasting was enabled first on the PVC which is set up between NPS and MCOE because NPS and MCOE had sufficient technological support. MCOE have bought a new router. This router was shipped with the latest release of Cisco IOS and had sufficient hardware support. Meanwhile 4 Mbytes of additional code memory and the Cisco IOS Release 11.1(1) was installed on the NPS Frame Relay router. After the routers were reconfigured for multicasting as discussed in Appendix B, and the MBone software was installed on one of the workstations at MCOE, a Sparc-20 on which Solaris 2.5 operating system was running, NPS and MCOE were ready for testing.

First, the new configuration for multicast needed to be tested. Before attempting to transmit multicast packets, it needed to be ensured that the new configuration was working properly. Also it needed to be ensured that multicast connectivity was established between these two sites. Multicast monitoring tools *mrinfo* and *mtrace* were used for testing.

The Cisco routers are able to respond to the *mrinfo* queries. After MCOE Frame Relay router was reconfigured, an *mrinfo* query-message was sent to it. The MCOE Frame Relay router reported that PIM was enabled and a multicast connection was established to the NPS Frame Relay router. This verified that the new configuration worked. *mtrace* was used to see the path taken by the multicast packets. This tool was run on the NPS Frame Relay router. The multicast packets were reaching to MCOE via NPS Frame Relay router and over the Frame Relay PVC set up between NPS and MCOE. This test was particularly important to rule out an alternate routing path that was theoretically possible via respective ISPs BBN Planet and CSUNet.

The results obtained from *mrinfo* and *mtrace* queries ensured that the multicast connectivity between NPS and MCOE had been successfully established. The next step was to test whether the MBone was possible on Frame Relay connections.

To ensure that *sd* sessions were announced from one router to another, a test *sd* session (named Test for MCOE) was created at NPS. The *sd* listener option in Cisco routers is used to monitor multicast connectivity. The routers on which the *sd* listener option is enabled can cache *sd* announcements. This cache file can be queried at any time to get information about any specific *sd* session. Since the *sd* listener option was enabled on both routers, the Test for MCOE session was expected to be cached by both routers. The "show ip sd" command is used to list the cached *sd* sessions. When this command was issued on the NPS Frame Relay router, the Test for MCOE session was listed. This ensured that the NPS Frame Relay router was able to talk to local hosts at

NPS. Thus host query/report messages were sent back and forth properly. The same command was then issued for the MCOE Frame Relay router. The Test for MCOE session was also cached by the MCOE Frame Relay router. This verified that the *sd* sessions were announced from one router to another over Frame Relay connections. After that, a live audio and video transmission was started by NPS. 15-16 frames per second were transmitted on average. The bit rate was 106Kbps. It was reported by MCOE that the video was received without any losses. Video transmitted by MCOE was received by NPS also without any losses. Thus video can be transmitted over Frame Relay connections with a quality near to the quality obtained over regular Internet connections. The audio was mostly understandable but sometimes choppy.

Similar tests were repeated for several times. Each time similar results were obtained. It is clear that live audio and video transmission over Frame Relay connections are possible.

C. PHASE II: TESTING LOW-COST PERSONAL COMPUTERS FOR MBONE

After it was ensured that multicast and MBone were possible over Frame Relay connections, the performance of MBone software on low-cost personal computers was evaluated. This was needed because most of the Monterey BayNet sites had Windows and Macintosh platforms. Without evaluating the performance of the MBone software on these platforms, it could not be concluded that the MBone can be practically implemented over the Monterey BayNet and used for distance learning.

The MBone software is available for Windows platforms. This software was downloaded and installed on one of the PCs which had Windows 95 running at NPS. The PC had the configuration listed in Figure 5.13. *sd/sdr* sessions created at MCOE were displayed by the *sd/sdr* running on the NPS PC. *sd/sdr* allowed us to join these sessions on the NPS PC. Audio was received and sent successfully. Again, the audio was mostly understandable but sometimes choppy. The video received was with the same quality that was received by UNIX-based workstations. Since the PC did not have a frame-grabber card and the Windows version of *vic* did not allow to send video at the time of this writing, the PC could not be tested for video transmission. In related testing, using the robust audio tool *rat* with redundancy (i.e. forward error correction) enabled, audio quality improves significantly and is intelligible even under conditions of 10% packet loss. We expect that use of *sdr, vic* (H.261 encoding) and *rat* (PCM redundancy)

are the optimum MBone software tools for use over Frame Relay. Further testing (especially during periods of congestion) will be useful.

D. SUMMARY

These experimental results show that multicast data delivery can be achieved on the Frame Relay connections. The implementation of the Internet Protocol (IP) multicast over Frame Relay PVCs using PIM between Cisco routers provides a multicast service which does not rely on dedicated multicast servers. The deployment of multicast enables live audio/video over Frame Relay connections which can be used for distance learning. Video and audio can be received and sent over Frame Relay connections with the same high quality that is usually provided over high-speed (1.536Mbps) Internet connections. Existing Mbone software that is available for low-cost Windows-based personal computers provides live audio/video to Windows platforms with the same quality that it provides to UNIX-based platforms.

As shown in these experiments, live audio and video delivery which can be used for distance learning is possible over Monterey BayNet Frame Relay links. These successes are of significant value to regional networked ocean science and education efforts.

VIII. CONCLUSIONS AND RECOMMENDATIONS

A. INTRODUCTION

This chapter summarizes the results obtained at the end of this thesis research. The conclusions and recommendations for future work are provided in this chapter.

B. CONCLUSIONS

Multicasting and the MBone are possible over Monterey BayNet which is a Frame Relay Wide-Area Network (WAN). This is achieved by implementing IP multicasting on already-installed Frame-Relay-capable Cisco routers. As discussed in earlier chapters, Frame Relay is a connection-oriented protocol which requires deployment of special multicast servers within the Frame Relay cloud. Enabling sparse-mode PIM (Protocol Independent Multicast) protocol (which is implemented by Cisco routers) eliminates the need for such servers. The implementation of native IP multicasting provides a multicast service model which is independent of servers and service providers.

The technology that Monterey BayNet sites have is sufficient to support this multicast service model. The implementation of multicast does not require buying any new equipment. The only requirement is to upgrade the router software and to install additional code memory. The built-in multicast support of Cisco routers provide the multicast connectivity over Monterey BayNet. The Cisco 2500 family of routers was originally selected for use of Monterey BayNet sites by Monterey BayNet net design team. This thesis research shows that the net design team made the right decision by selecting the Cisco routers. For future sites, Cisco routers continue to be the right selection because they provide a working solution for multicast that is compatible regionally.

Implementing IP multicast over Monterey BayNet enables live audio and video transmission by using MBone software tools over Frame Relay links. The video can be transmitted over Frame Relay links with a quality similar to that transmitted over regular Internet connections. The audio is problematic as always. It is understandable but sometimes choppy. The use of *rat*'s redundancy control mechanism (i.e. forward error correction) significantly improves

the quality of the audio. We expect that use of *sdr, vic* (H.261 encoding) and *rat* (PCM redundancy) are the optimum MBone software tools for use over Frame Relay.

The MBone software tools are available for Windows platforms. The experimental results of this thesis research also show that MBone software tools provide similar performance on both UNIX platforms and low-cost Windows 95 based personal computers (PCs). These results ensure that MBone can be practically implemented over Monterey BayNet and used for distance learning. Monterey BayNet sites can now take full advantage of functionality provided by today's Internet.

Without network monitoring for problem diagnosis, continuous multicast packet delivery cannot be achieved over Monterey BayNet links. The multicast monitoring pages developed in this thesis research provide effective, interactive, portable and free software monitoring tools to ensure that multicast traffic problems can be diagnosed and corrected. This was previously impossible because public domain multicast monitoring tools are only available on UNIX-based platforms. By using the Web and scripting languages, these tools are made available for use by Monterey BayNet school sites that mostly have Windows and Macintosh platforms.

C. RECOMMENDATIONS FOR FUTURE WORK

Pacific Bell CalREN grants expires on October 1, 1996. After that time, the Monterey BayNet sites need to pay for the network connectivity. Because of the limited budget available to Monterey BayNet sites, some participating sites might want to decrease the line capacities to 56 Kbps. However a good-quality of video and audio transmission requires use of 128 Kbps line. The performance of MBone is low on 56 Kbps links. For sites that want to join the MBone, it is highly recommended to have at least 128 Kbps lines. If the budget permits, upgrading to 256 Kbps is worth consideration.

Performance of multicasting and MBone on low-speed Frame-Relay connections and low-cost personal computers was evaluated during several experiments. These tests were performed only between three sites. We still do not know what will be the performance of Mbone and multicasting over Monterey BayNet for ongoing events with many participating Monterey BayNet sites. Future work definitely includes testing multiple Frame Relay sites simultaneously and attempting production use of the Monterey BayNet MBone in support of the Monterey Bay

Aquarium's daily Bay Link educational program.

As discussed in Chapter V, firewalls need to be configured properly for multicast traffic. Otherwise they block the multicast traffic. The multicast monitoring tools introduced in Chapter VI work on an NPS server which is behind a firewall. However the NPS firewall cannot be properly configured for multicasting (especially for IGMP which is essential for multicast monitoring) due to an inadequate firewall implementation. This restriction prevents proper IGMP traffic routing and thus proper multicast monitoring. *mtrace* and *mrinfo* gateways cannot be used to monitor the multicast traffic outside the inadequate firewall. An important area for future work is to upgrade firewall software and enable IGMP traffic.

Asynchronous Transfer Mode (ATM) is an other ongoing research area at NPS. During our work on multicast monitoring, we realized that monitoring multicast connectivity was also an important issue for ATM networks. We considered reimplementing the Frame Relay work produced here for the NPS ATM LAN. However ATM is still not ready (Courtney, 1996). There are still lots of problems that need to be resolved. At that time, what we have produced for Frame Relay networks can be ported to the NPS ATM LAN. Such work assumes the implementation of IP over ATM. The current NPS ATM LAN is based on Permanent Virtual Connections (PVCs). Therefore manual or automatic Address Resolution Protocol (ARP) configuration will be needed. Additionally use of ATM ARP servers makes this configuration easier. The Cisco 7000 router used by the NPS Church Computer Center can be configured as an ATM ARP server since the required hardware and software to support such a service is already in hand. (Cisco, 1996) Once any Switched Virtual Connections (SVCs) are set up and the Cisco 7000 router is configured as the ATM ARP server, IP multicasting may be implemented and these multicast monitoring tools may be ported to the NPS ATM LAN.

Multicast scope control is essential in order to safely integrate regional Monterey BayNet Frame Relay MBone with the global MBone. Both the regular MBone scope controlling mechanism (TTL) and our proposed solution are user-dependent. This is because the current implementation of *mrouted* does not provide any other scope control mechanism (such as administratively decremented TTL or address filtering). As discussed in Chapter V, decrementing TTL to an administratively controlled value might make scope controlling easier. This is still an option for future work and worth consideration. The global Mbone community needs to be convinced that such a mechanism is needed and useful. The work presented here provides strong

evidence that further controls are necessary.

D. SUMMARY

Multicasting and the MBone are possible over the Monterey BayNet. This enables live audio/video by using MBone software tools over Frame Relay links. The current MBone software tools provide the same performance that they provide on regular Internet connections even on low-speed Frame Relay connections and low-cost personal computers. Implementation of multicast requires monitoring of multicast connectivity. Multicast monitoring tools are developed to meet the monitoring needs of the Monterey BayNet sites. They are now able to monitor the regional multicast connectivity.

Performance of MBone is slow on 56 Kbps links. 128 Kbps line capacity is the minimum requirement for multicast connectivity. If the budget permits it is worth consideration to upgrade the line capacity to 256 Kbps. We still do not know the performance of the Frame Relay MBone with multiple participating sites. The performance of MBone need to be evaluated with multiple participating Monterey BayNet sites. For multicast scope controlling, mechanisms other than use of TTL need to be examined. Otherwise, scope controlling will always be user dependent. Finally, due to inadequate software implementation, the NPS firewall cannot be configured for multicasting properly. This problem prevents IGMP traffic and thus effective monitoring. The firewall software needs to be upgraded in order to enable IGMP.

We live in information age. people are willing to have information when and where they need it. Implementation of multicasting over Monterey BayNet helps meet these growing needs of Monterey BayNet sites.

APPENDIX A. IANA INTERNET MULTICAST ADDRESSES

Multicast addresses are used to distinguish host groups from each other. Each host group has a unique group address or multicast address. IP multicast uses Class D IP addresses ranging from 224.0.0.0 through 239.255.255.255. IP multicast addresses are administratively controlled by Internet Assigned Numbers Authority (IANA). Some IP addresses are administratively assigned for use by permanent groups (called "well-known addresses"). IP multicast addresses other than these well-known IP multicast addresses can be used for transient groups. Current IP multicast addresses are listed below. Note that the range of addresses between 224.0.0.0 and 224.0.0.255, inclusive, is reserved for the use of routing protocols and other low-level topology discovery or maintenance protocols (such as gateway discovery and group membership reporting). Multicast routers should not forward any multicast datagram with destination addresses in this range, regardless of its TTL. The current version of this list can be found at

ftp://venera.isi.edu/in-notes/iana/assignments/multicast-addresses

```
224.0.0.0  Base Address (Reserved)                  [RFC1112,JBP]
224.0.0.1  All Systems on this Subnet               [RFC1112,JBP]
224.0.0.2  All Routers on this Subnet               [JBP]
224.0.0.3  Unassigned                               [JBP]
224.0.0.4  DVMRP    Routers                         [RFC1075,JBP]
224.0.0.5  OSPFIGP  OSPFIGP All Routers             [RFC1583,JXM1]
224.0.0.6  OSPFIGP  OSPFIGP Designated Routers      [RFC1583,JXM1]
224.0.0.7  ST Routers                               [RFC1190,KS14]
224.0.0.8  ST Hosts                                 [RFC1190,KS14]
224.0.0.9  RIP2 Routers                             [RFC1723,GSM11]
224.0.0.10 IGRP Routers                             [Farinacci]
224.0.0.11 Mobile-Agents                            [Bill Simpson]
224.0.0.12 DHCP Server / Relay Agent                [RFC1884]
224.0.0.13 All PIM Routers                          [Farinacci]
224.0.0.14 RSVP-ENCAPSULATION                       [Braden]
224.0.0.15-224.0.0.255 Unassigned                   [JBP]
224.0.1.0  VMTP Managers Group                      [RFC1045,DRC3]
224.0.1.1  NTP      Network Time Protocol           [RFC1119,DLM1]
224.0.1.2  SGI-Dogfight                             [AXC]
224.0.1.3  Rwhod                                    [SXD]
224.0.1.4  VNP                                      [DRC3]
224.0.1.5  Artificial Horizons - Aviator            [BXF]
224.0.1.6  NSS - Name Service Server                [BXS2]
224.0.1.7  AUDIONEWS - Audio News Multicast         [MXF2]
224.0.1.8  SUN NIS+ Information Service             [CXM3]
224.0.1.9  MTP Multicast Transport Protocol         [SXA]
```

```
224.0.1.10 IETF-1-LOW-AUDIO                          [SC3]
224.0.1.11 IETF-1-AUDIO                              [SC3]
224.0.1.12 IETF-1-VIDEO                              [SC3]
224.0.1.13 IETF-2-LOW-AUDIO                          [SC3]
224.0.1.14 IETF-2-AUDIO                              [SC3]
224.0.1.15 IETF-2-VIDEO                              [SC3]
224.0.1.16 MUSIC-SERVICE                             [Guido van Rossum]
224.0.1.17 SEANET-TELEMETRY                          [Andrew Maffei]
224.0.1.18 SEANET-IMAGE                              [Andrew Maffei]
224.0.1.19 MLOADD                                    [Braden]
224.0.1.20 any private experiment                    [JBP]
224.0.1.21 DVMRP on MOSPF                            [John Moy]
224.0.1.22 SVRLOC                                    [Veizades]
224.0.1.23 XINGTV                                    <hgxing@aol.com>
224.0.1.24 microsoft-ds                              <arnoldm@microsoft.com>
224.0.1.25 nbc-pro                                   <bloomer@birch.crd.ge.com>
224.0.1.26 nbc-pfn                                   <bloomer@birch.crd.ge.com>
224.0.1.27 lmsc-calren-1                             [Uang]
224.0.1.28 lmsc-calren-2                             [Uang]
224.0.1.29 lmsc-calren-3                             [Uang]
224.0.1.30 lmsc-calren-4                             [Uang]
224.0.1.31 ampr-info                                 [Janssen]
224.0.1.32 mtrace                                    [Casner]
224.0.1.33 RSVP-encap-1                              [Braden]
224.0.1.34 RSVP-encap-2                              [Braden]
224.0.1.35 SVRLOC-DA                                 [Veizades]
224.0.1.36 rln-server                                [Kean]
224.0.1.37 proshare-mc                               [Lewis]
224.0.1.38 dantz                                     [Yackle]
224.0.1.39 cisco-rp-announce                         [Farinacci]
224.0.1.40 cisco-rp-discovery                        [Farinacci]
224.0.1.41 gatekeeper                                [Toga]
224.0.1.42 iberiagames                               [Marocho]
224.0.1.43-224.0.1.255  Unassigned                   [JBP]
224.0.2.1  "rwho" Group (BSD) (unofficial)           [JBP]
224.0.2.2  SUN RPC PMAPPROC_CALLIT                   [BXE1]
224.0.2.064-224.0.2.095 SAIC MDD Service             [Bressler]
224.0.3.000-224.0.3.255 RFE Generic Service          [DXS3]
224.0.4.000-224.0.4.255 RFE Individual Conferences   [DXS3]
224.0.5.000-224.0.5.127 CDPD Groups                  [Bob Brenner]
224.0.5.128-224.0.5.255 Unassigned                   [IANA]
224.0.6.000-224.0.6.127 Cornell ISIS Project         [Tim Clark]
224.0.6.128-224.0.6.255 Unassigned                   [IANA]
224.0.7.000-224.0.7.255 Where-Are-You                [Simpson]
224.0.8.000-224.0.8.255 INTV                         [Tynan]
224.0.9.000-224.0.9.255 Internet Railroad            [Malamud]
224.0.10.000-224.0.10.255 DLSw Groups                [Lee]
224.1.0.0-224.1.255.255 ST Multicast Groups          [RFC1190,KS14]
224.2.0.0-224.2.255.255 Multimedia Conference Calls  [SC3]
224.252.0.0-224.255.255.255 DIS transient groups     [Joel Snyder]
232.0.0.0-232.255.255.255  VMTP transient groups     [RFC1045,DRC3]
```

APPENDIX B. ROUTER CONFIGURATIONS FOR MULTICASTING

Chapter V described the considerations about enabling multicasting over the Monterey BayNet. The first configuration script provided below is an example for Monterey BayNet sites planning to join MBone. The second script is the configuration script used for the NPS Frame Relay router.

The conventions throughout the scripts are as follows. Plain texts are the ones generated by the router. **Boldface** texts indicate user input. This information must be provided by the user. The author's comments are given after // (double slashes) and in *italics*. These are only comments. The user need not type them during the configuration process.

A. ROUTER CONFIGURATION FOR MONTEREY BAYNET SITES

This section provides the router configuration for any Monterey BayNet site. It is assumed that you have a router of Cisco 2500 family and this router is already configured for Frame Relay connectivity. If you need to configure the router for Frame Relay connectivity also, please first refer to (Bigelow, 1995). It provides a full configuration script for Frame Relay configuration.

As disscussed in Chapter V, upgrading router software and hardware is essential for Monterey BayNet sites. Before attempting to configure the router for multicast, ensure that you have 4 MBytes of code memeory, 2 MBytes of main memory, latest Boot ROM level (currently it is 10.2(8a)) and the latest release of Cisco IOS (currently it is 11.1(3)). The router configured as an example was an Cisco 2503 and had 4 MBytes of code memeory, 2Mbytes of main memory, Boot ROM Level 10.2(8a) and Cisco IOS Release 11.1(1).

First enable the multicast routing feature on your router. This must be done in the global router configuration mode.

```
dolphin>enable
Password:                    // enter the enable mode password
dolphin#config terminal
Enter configuration commands,one per line.End with CNTL/Z.
dolphin(config)#ip muticast-routing
```

Enable sparse-mode PIM on the ethernet interface. Sparse-mode PIM requires a Rendezvous Point. For the sites for which the Internet access is provided via MCOE, the MCOE Frame Relay router (205.155.43.1) is the RP. The sites which the Internet access is provided directly via CSUMB, the CSUMB Frame Relay router (137.145.176.1) is the RP. Monterey County topology is shown in Figure 5.12. For monitoring purposes, enabling the *sd* listener option is highly recommended.

```
dolphin(config)#interface ethernet 0
dolphin(config-subif)#ip pim sparse-mode
dolphin(config-subif)#ip pim rp-address 205.155.43.1
dolphin(config-subif)#ip sd listen
```

Multicasting must be enabled for the serial interface also. The *sd* listener option must be applied in the interface configuration mode. First enable the *sd* listener option on the serial interface before enabling multicasting on the logical interface.

```
dolphin(config-subif)#interface serial 0
dolphin(config-if)#ip sd listen
```

The multicast topology should reflect the regular IP topology. If there are two PVCs configured for your site, enable multicasting only on the interface that you use to access Internet services (such as ftp and http). This is the case for the sites having PVCs to both MCOE and CSUMB. If your site is one of them, enable multicasting only on the interface that connects you to CSUMB. Otherwise just enable multicasting on the single PVC configured. Again, the rendezvous point is an important issue. Use the same RP defined for the Ethernet interface.

```
dolphin(config-subif)#interface serial 0.2
dolphin(config-subif)#ip pim sparse-mode
dolphin(config-subif)#ip pim rp-address 205.155.43.1
```

Controlling the scope of the multicast traffic is an important issue. TTL is one of the efficient methods used for scope controlling. Only packets that have a TTL value greater than the ttl-threshold value of the interface can be forwarded over that interface. For most Monterey BayNet sites, 8 is the TTL value applied to the multicast enabled interfaces. See Figure 5.12 for the TTL value that should be applied to your site.

```
dolphin(config-subif)#ip multicast ttl-threshold 8
```

The complete configuration script is repeated in the following.

```
dolphin>enable
Password:
dolphin#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
dolphin(config)#ip multicast-routing
dolphin(config)#interface ethernet 0
dolphin(config-subif)#ip pim sparse-mode
dolphin(config-subif)#ip pim rp-address 205.155.43.1
dolphin(config-subif)#ip sd listen
dolphin(config-subif)#interface serial 0
dolphin(config-if)#ip sd listen
dolphin(config-subif)#interface serial 0.2
dolphin(config-subif)#ip pim sparse-mode
dolphin(config-subif)#ip pim rp-address 205.155.43.1
dolphin(config-subif)#ip multicast ttl-threshold 8
dolphin(config-if)#^Z


dolphin#write terminal // this is issued to see the current configuration
Building Configuration...
Current configuration:
!
version 11.1
service udp-small-servers
service tcp-small-servers
!
hostname dolphin
!
enable password xxxxxxx

ip multicast-routing
!
interface Ethernet0
 ip address 131.120.211.50 255.255.255.0
 ip pim sparse-mode
 ip sd listen
!
interface Serial0
 no ip address
 ip sd listen
 encapsulation frame-relay
 no ip route-cache
 frame-relay lmi-type ansi
```

```
!
interface Serial0.1 point-to-point
 description NPS-CSUMB DLCI 802
 ip address 198.189.239.90 255.255.255.252
 frame-relay interface-dlci 802 broadcast
!
interface Serial0.2 point-to-point
 description NPS-MOCOE DLCI 902
 ip address 198.189.239.6 255.255.255.252
 ip pim sparse-mode
 ip multicast ttl-threshold 8
 frame-relay interface-dlci 902 broadcast
!
interface Serial1
 no ip address
 no ip route-cache
 shutdown
!
interface BRI0
 no ip address
 no ip route-cache
 shutdown
!
router igrp 11
 network 198.189.239.0
!
ip name-server 205.155.43.2
no ip classless
ip pim rp-address 205.155.43.1

!
line con 0
line aux 0
 transport input all
line vty 0 4
 password xxxxx
 login
!
end
```

B. ROUTER CONFIGURATION FOR THE NPS FRAME RELAY ROUTER

The NPS Frame Relay router `dolphin.nps.navy.mil` is a Cisco 2503. It has one Ethernet interface and two serial interfaces. There are two PVCs created for NPS: one to CSUMB on serial 0.1 and one to MCOE on serial 0.2. The multicasting is enabled only on the serial 0.2 interface. Release 11.1 (1) of the Cisco IOS has been installed and the boot ROM has been replaced (level 10.2(8a)). The router has 4 MBytes of code memory and 2 MBytes of main memory.

The configuration provided in the previous section is the basic configuration for each Monterey BayNet site. As discussed in Chapter V, the configuration of the NPS Frame Relay router should be different from other Monterey BayNet sites because it is a border router. In addition to the above configuration, the router must be configured so that

1. NPS is not in a position the Internet Service Provider (ISP) of Monterey BayNet, and
2. NPS is not the MBone provider of the Monterey BayNet

To prevent NPS from being the MBone provider of the Monterey BayNet, an access list is defined for multicast groups. The only multicast packets that can pass through the router are the packets with a multicast address of 224.0.1.20 which is reserved for "any private experiment" (Appendix A). The access lists are defined in global configuration mode. The access list defined for multicast groups is used for both Ethernet 0 and serial 0.2 interfaces.

```
dolphin(config)#access-list 1  permit  224.0.1.20 0.0.0.0
dolphin(config)#access-list 1  deny   any
```

The following command is issued to enforce to block multicast packets with a multicast address other than 224.0.1.20. It is applied on both Ethernet and serial 0.2 interfaces.

```
dolphin(config-subif)#ip igmp access-group 1
```

The NPS Frame Relay router should thus only let regional multicast packets pass through. For regional multicast traffic, the TTL value should be between 16 and 32. Multicast packets are

not forwarded over either Ethernet 0 or serial 0.2 interfaces unless they have a TTL value greater than 16. The TTL-threshold value for both interfaces is 16.

```
dolphin(config-subif)#ip multicast ttl-threshold 16
```

The NPS Frame Relay router is directly connected to an mrouter running DVMRP (`betelguise.cs.nps.navy.mil`). Cisco IOS does not directly support DVMRP. However, PIM-enabled Cisco routers can interoperate with the router running DVMRP and can dynamically discover DVMRP routers on attached networks. Once the DVMRP router has been discovered, the router periodically transmits DVMRP report messages advertising the reachable unicast sources in the PIM domain. (Cisco, 1996) The command `ip dvmrp metric` is used to configure what sources are advertised and what metrics are used while unicast sources are advertised. The NPS Frame Relay router can advertise only the sources within the Monterey BayNet (205.155.0.0) and the metric 1 is used. The access list 2 is defined for the sources that can be advertised.

```
dolphin(config)#access-list 2 permit 205.155.0.0  0.0.255.255
dolphin(config)#access-list 2 deny   any
dolphin(config-subif)#ip dvmrp metric 1 list 2
```

As discussed in Chapter V, the NPS Frame Relay router is connected to one of the multicast- capable subnetworks of NPS. Router Information Protocol (RIP) is used as the IP routing protocol within the NPS network. However, Internet Gateway Routing Protocol (IGRP) is used for the Monterey BayNet. The RIP information needs to be redistributed to IGRP domain (Cisco, 1996). The following commands are applied in router configuration mode:

```
dolphin(config-subif)#router igrp 11
dolphin(config-router)#network 198.189.239.0
dolphin(config-router)#redistribute rip
```

The full configuration script is the following.

```
dolphin>enable
Password:
```

```
dolphin#config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
dolphin(config)#ip muticast-routing
dolphin(config)#access-list 1  permit 224.0.1.20   0.0.0.0
dolphin(config)#access-list 1  deny    any
dolphin(config)#access-list 2 permit 205.155.0.0   0.0.255.255
dolphin(config)#access-list 2 deny     any
dolphin(config)#interface ethernet 0
dolphin(config-subif)#ip pim sparse-mode
dolphin(config-subif)#ip pim rp-address 205.155.43.1
dolphin(config-subif)#ip sd listen
dolphin(config-subif)#ip multicast ttl-threshold 16
dolphin(config-subif)#ip dvmrp metric 1 list 2
dolphin(config-subif)#ip igmp access-group 1
dolphin(config)#interface serial 0
dolphin(config-if)#ip sd listen
dolphin(config-subif)#interface serial 0.2
dolphin(config-subif)#ip pim sparse-mode
dolphin(config-subif)#ip pim rp-address 205.155.43.1
dolphin(config-subif)#ip multicast ttl-threshold 16
dolphin(config-subif)#ip igmp access-group 1
dolphin(config-subif)#router igrp 11
dolphin(config-router)#network 198.189.239.0
dolphin(config-router)#redistribute rip
dolphin(config-if)#^Z

dolphin#write terminal   // this is issued to see the current configuration
Building configuration...

Current configuration:
!
version 11.1
service udp-small-servers
service tcp-small-servers
!
hostname dolphin
!
enable password xxxxxxx
!
ip multicast-routing
!
interface Ethernet0
 ip address 131.120.211.50 255.255.255.0
 ip pim dense-mode
 ip multicast ttl-threshold 16
 ip igmp access-group 1
```

```
 ip dvmrp metric 1 list 2
 ip sd listen
!
interface Serial0
 no ip address
ip sd listen
 encapsulation frame-relay
 no ip route-cache
 frame-relay lmi-type ansi
!
interface Serial0.1 point-to-point
 description NPS-CSUMB DLCI 802
 ip address 198.189.239.90 255.255.255.252
 frame-relay interface-dlci 802 broadcast
!
interface Serial0.2 point-to-point
 description NPS-MOCOE DLCI 902
 ip address 198.189.239.6 255.255.255.252
 ip pim dense-mode
 ip multicast ttl-threshold 16
 ip igmp access-group 1
 frame-relay interface-dlci 902 broadcast
!
interface Serial1
 no ip address
 no ip route-cache
 shutdown
!
interface BRI0
 no ip address
 no ip route-cache
 shutdown
!
router rip
 network 131.120.0.0
!
router igrp 11
 redistribute rip
 network 198.189.239.0
!
ip name-server 205.155.43.2
ip name-server 131.120.254.58
no ip classless
access-list 1 permit 224.0.1.20
access-list 1 deny   any
access-list 2 permit 205.155.0.0 0.0.255.255
```

```
access-list 2 deny   any
snmp-server community public RO
snmp-server community private RW
snmp-server contact Roy Romo romo@nps.navy.mil
!
line con 0
line aux 0
 transport input all
line vty 0 4
 password xxxxxx
 login
!
end
```

APPENDIX C. MBONE CONFIGURATION FOR WINDOWS PLATFORMS

MBone software makes live audio and video transmission possible across the Internet. This software is available for almost every platform (except Macintosh, currently). Installation and use of this software is straightforward. The following subsections describe the installation process of MBone software available for Windows platforms (Windows 95). This software is free and comes without any guarantee. Most of the software is alpha/beta releases but is well-tested, secure and reliable. Before attempting to install and use these tools, please read the README files coming with the software itself and the on-line documentation available via the World Wide Web.

A. OBTAINING THE SOFTWARE

Not all MBone software is available for Windows platforms. The following is the list of MBone tools available for Windows platforms at the time of this writing. The list reflects the latest version of the software and the pointers to download sites. A complete archive of available tools can be found at *ftp://cs.ucl.ac.uk/mice/videoconference*

sdr (Session Directory): It is a tool based on the LBL session directory (*sd*). It is also used to create and announce MBone sessions and launch other MBone tools. *sdr* and *sd* are not compatible.
File Name: Winsdr22.zip
Download Site: *ftp://cs.ucl.ac.uk/mice/videoconference/sdr/*

sd (LBL Session Directory): It is used to create and announce MBone sessions. It is also used to launch the other MBone tools.
File Name: SD-Win32-2.0a3.zip
Download Site: *http://archive.thepoint.net/Win32SD/*

rat (Robust Audio Tool): Rat is the other audio tool available for Windows platforms. The redundancy control mechanism makes it more robust than *vat* when Frame Relay packet loses

occurs

File Names: rat-2.6a4-Win32.zip

Win95dll.zip

Download Site: *ftp://cs.ucl.ac.uk/mice/videoconference/rat/*

vat (Visual Audio Tool): Vat is used to send and receive audio.

File Name: vatbin-4.0b2-win95.zip

Download Site: *ftp://ftp.ee.lbl.gov/conferencing/vat/alpha-test/*

vic (Video Conferencing Tool): vic is normally used to send and receive video. However, the Windows 95 version of *vic* is not yet capable of sending video.

File Name: vicbin-2.7b3-win95.zip

Download Site: *ftp://ftp.ee.lbl.gov/conferencing/vic/2.7/*

nvat (Network Video Audio Tool): Nvat is a combination of audio and video tools. It is compatible with vat and nv (a video tool which is not available for Windows platforms).

File Names: nvatrecv.zip (receive only)

nvatsend.zip (send only)

Download Site: *ftp://ftp.meshnet.or.jp/pub/mesh/nvat*

B. INSTALLING THE MBONE SOFTWARE

All MBone software comes in compressed forms, a .zip extension is used for compressed files. These files can be unzipped by using any of the decompression tools available for Windows platforms. Be sure that these files are unzipped in an empty directory. Before installing this software, create a new directory or folder under your home directory which is C:\ for most people and name it "mbone". This folder will be used to keep the executable files. This is not a requirement to be able to run the MBone software. It is recommended only to keep track of the MBone software in your system.

1. Installing sdr

Winsdr22.zip is a whole package that includes the precompiled binaries of *vat* and *vic* as well as *sdr*. Two supporting programs used by *sdr, cal* and *rm*, are also included in this package. When this file is unzipped, the files are extracted into a directory named *mbone*. This directory is created under the directory where the compressed file was unzipped. This directory also has a subdirectory named *sdr*. It contains the *sdr* configuration file *prefs* and subdirectories *cache* to cache the session announcements and *plugins* to put locally defined media descriptions. When you run *sdr*, it looks for the *sdr* directory under your home directory, which is again *C:* for most users. The *sdr* directory should be moved to the home directory. The *sdr.exe, vat.exe,* and *vic.exe* files should be moved under the directory that you have created to keep MBone executable files (*C:\mbone*).

2. Installing sd

After you have downloaded the file named SD-Win32-1.4a.zip, unpack this file in an empty directory. Decompression of this file yields the files *sd.exe, mfc40.dll, msvcrt.dll* and *readme* files. Copy the *sd.exe* file into the mbone directory that was created. Dynamic Link Library (dll) files should be moved to the *C:\windows\system* directory. Some of the dll files are missing from the standard distribution of Windows 95. The dll files provided in this package are two of these files.

3. Installing rat

The download site listed for *rat* above is the official site for *rat*. *rat* is designed and developed at this site. This site always has the most up-to-date version of *rat*. When the compressed file is unzipped, it will be the precompiled binary version of *rat*. To run this tool, it is necessary to move the executable file under *C:\mbone* directory. It is also necessary to download the *Win95dll.zip* file. It contains two *dll* files required by *rat*: *tcl75.dll* and *tk41.dll*. After unzipping the file, move these two dll files to *C:\windows\system*.

4. Installing vat

Moving the executable file obtained by unzipping the compressed file to the *C:\mbone* directory is the only requirement for *vat*. If is already installed *sdr*, *vat* is already available. It is not necessary to download and install this packet. However, ensure that the latest version of *vat* is installed. The download site listed previously for *vat* is the official site. The most up-to-date version of *vat* can be found at this site.

5. Installing vic

If is already installed *sdr*, it is probably not necessary to install *vic*. Again, ensure that the latest version is installed. The download site listed for *vic* is the official site. The latest version is available at this site. If is not installed *sdr*, after downloading the related compressed file, unzip the file. Move the resulting executable file to *C:\Mbone*.

6. Installing nvat

nvat is a tool which includes both audio and video tools. It has two portions: one for receiving and one for sending. Unless a video capture card is installed in the computer, which is needed for video transmission, it is not necessary to download and install the send portion of *nvat*. After the files are unzipped, copy the resulting executable files under the *C:\mbone* directory. The performance of *nvat* is slow with less than 32Mbytes of RAM. Therefore, installation of this tool is not recommended.

C. RUNNING MBONE SOFTWARE

Windows 95 compatible MBone software tools are 32-bit applications. They are written for Windows 95 and Windows NT platforms and the TCP/IP stack needs to be installed and run. Before attempting to use these tools, the Windows 95 TCP/IP stack needs to be properly installed and set up. The details for Window 95 TCP/IP internetworking and TCP/IP setup are given in (Microsoft, 1995). It may also be necessary to upgrade the network driver software. Older

versions of driver software often do not support multicast. Therefore, unless a new network card is installed, the driver software may need to be upgraded. The best way to upgrade driver software is to check the Web pages of the company making your network interface card.

sd and *sdr* can be run either by double clicking on the *sd/sdr* icons or from the command line. *sd/sdr* are used for launching the other MBone tools as well as announcing and creating MBone sessions. Current version of *sd* does not allow new sessions to be created. To be able to run other MBone tools via *sd/sdr*, the binary version of these tools need to reside in a directory which is in the correct path. The PATH environment variable should point to this directory. If this is not the case, updating the PATH is necessary. The easiest way to do this is to modify the autoexec.bat file. Edit the autoexec.bat file by using any text editor, such as *notepad* or *edit*, and add "*C:\MBONE*" to the line that defines the environment variable PATH. The following is a sample of a PATH definition in an *autoexec.bat* file.

PATH = C:\; C:\WINDOWS;C:\COMMANDS;C:\MBONE

After modifying the PATH, it is necessary to run autoexec.bat file manually or re-boot the system. Thus *vic*, *vat*, *rat* can be run via *sd/sdr*.

vic, *vat*, and *rat* can be also run from the command line. To be able to run these tools from the command line, click on the start button of Windows 95 and select *run* (*start -> run*). Type the path that the executable files reside in and the name of the executable file. The address of the multicast group to be joined and the port number need to be provided as command line arguments. If the PATH variable has already been modified, the whole path that the executable file can be found does not need to be typed. Just type the name of the executable file and multicast group address and the port number. Figure C.1 shows how *vat*, *rat*, and *vic* can be run from the command line assuming that the PATH has been modified.

```
c:\vat  224.2.117.196/30089

c:\rat  224.2.117.196/30089

c:\vic 224.2.117.196/30089
```

Figure C.1 Command Line Invocation of *vat*, *rat*, and *vic*

nvat can be run either by double clicking on the *nvat* icon or from the command line. If run from the command line, be sure that the executable of *nvat* resides in a directory pointed by the

environment variable PATH. When *nvat* is run, a window asking for the multicast group address and the port number will pop up. Provide the address of the multicast group to be joined and the port number. Since the PC that was used to test the MBone tools did not have a video capture card, the send portion of *nvat* could not be tested.

D. DOCUMENTATION FOR MBONE SOFTWARE

The packages used for installation do not include any type of user information. README files that are included within the packages give only brief information about the installation of the software. However, extensive on-line information is available via the World Wide Web from many sites. When a net search is done a lot of pointers to this on-line information can be found. The following is the list of sites that provide on-line information about both installation and use of MBone software. These sites do not provide platform-specific information, but provide general guidelines for the tools.

sdr Installation Manual:	*http://www-mice-nsc.cs.ucl.ac.uk/mice-nsc/tools/install-sdr.html*
sdr User Manual:	*http://ugwww.ucs.ed.ac.uk/mice/archive/sdr.html*
sd User Manual:	*http://www-mice-nsc.cs.ucl.ac.uk/mice-nsc/tools/user-sd.html*
sd Installation Manual:	*http://www-mice-nsc.cs.ucl.ac.uk/mice-nsc/tools/install-sd.html*
rat Installation Manual:	*http://www-mice-nsc.cs.ucl.ac.uk/mice-nsc/tools/install-rat.html*
rat General Guidelines:	*http://boom.cs.ucl.ac.uk/mice/rat/*
vat Installation Manual:	*http://www-mice-nsc.cs.ucl.ac.uk/mice-nsc/tools/install-vat.html*
vat User Manual:	*http://www-mice-nsc.cs.ucl.ac.uk/mice-nsc/tools/user-vat.html*
vic Installation Manual:	*http://www-nrg.ee.lbl.gov/vic/#installation*
vic General Guidelines:	*http://www-nrg.ee.lbl.gov/vic/*

In addition to the on-line information, (Emswiler, 1995) provides a user manual for MBone related software. The MBone software is discussed in greater detail in (Kumar, 1996).

APPENDIX D. REQUIRED CHANGES FOR THE USE OF MULTICAST MONITORING TOOLS

Since multicast monitoring tools need to create UDP sockets, they should be run by the super-user or the root. They create these sockets to send and receive IGMP control messages. All they do is send and receive control messages. Therefore, it is not a system security violation if they are run by regular users. As an administrator you can let either all users or a set of users use these tools. For world-wide accessible monitoring pages, you need to let all users (including *nobody*) to use these tools.

In order to let all users run these tools, the *setuid* bit of *mrinfo* and *mtrace* must be set. In the UNIX file system, each file has a set of twelve bits called mode bits. Nine of these are permission bits that control who can read, write, and execute the contents of the file. The other three bits affect the operation of the executable programs. *Setuid* is one of them and allows programs to access files and processes that would be otherwise be off limits to the user that runs them.(Nemeth, 1995) Giving execute permission to all users is not a sufficient solution. Since even execute permissions are given to all users, these tools are still required to be run by the root. The program itself is able to check its owner. If it is not owned by the root, the program terminates and gives the "must be root" warning message. Therefore the *setuid* bit must be set in order to allow anybody other than toot to run these tools. The owner of the file (root in our case) can set the mode bits by using the `chmod` command. By root issuing the following the *setuid* bit can be set for both *mrinfo* and *mtrace*.

```
#chmod u+s mtrace
#chmod u+s mrinfo
```

This change is sufficient to be able to run these tools by anybody (including Weeb-page remote user *nobody*). *mrinfo* and *mtrace* can be run via Web pages with these mode bit configurations. To view the permissions on *mtrace* and *mrinfo*, type the `ls -l` command. The permission bit settings on *mtrace* and *mrinfo* are below:

```
#ls -l mtrace
-rwsr-xr-x   1 root root  123183 Jul 30 18:45 mtrace*
#ls -l mrinfo
-rwsr-xr-x   1 root root   91165 Jul 30 18:48 mrinfo*
```

You may not always want everybody to able to run *mrinfo* and *mtrace*. This requires

new changes in addition to the one described above. In the second case where only a set of users are allowed to run *mrinfo* and *mtrace*, a new group must be defined for those users. The UNIX operating system provides a concept called ownership. The owner of the file is always a single user and has primary control of it. UNIX OS also allows many people to be the owner of a file as long as they are all part of a single UNIX group. The groups are created by the super-user or root and defined in `/etc/group` file. This file contains the group names and a list of each group's members. Each line in this file represents a group and has four fields. The general layout for a line in that file is as follows.

groupname:encypted password:GID Number:list of group members

Groupname must be up to 8 characters for ATT systems (there is no limit for BSD systems). The standard UNIX distributions do not provide group passwords although the field is allocated. A star (*) is put in this field conventionally. Group ID (*gid*) number must be an integer between 0 and 32767. GID 0 is reserved for the root. GID 1 is generally used for the group "daemon." User login names that will belong to that group are listed in the last field and separated by commas. The user login names must appear as in the `/etc/passwd` file. (Nemeth, 1995)

To create a new group, the `/etc/group` file must be edited by using one of the text editor programs such as *emacs* or *vi*. You ned to be the super-user to edit this file. At NPS, a group called *studRoot* is created for the set of users allowed to run multicast monitoring tools. The following is the line from the `/etc/group` file which defines the group *studRoot* and its members.

```
studRoot:*:170:erdogan,brutzman,edwardse,mttamer
```

After the `/etc/group` file has been edited, the group ownership of multicast monitoring tools (namely *mrinfo* and *mtrace)* needs to be changed. The `chgrp` command is used to change the group ownership of any file. Only the owner of a file or the super-user may change the group of that file.

```
#chgrp studRoot mtrace
#chgrp studRoot mrinfo
```

If you do not set *setuid* bit, these tools still cannot be run by anybody including *studRoot*. Therefore, in either case the setuid bit must be set. After the setuid bit is set as described above, these tools can be run by anybody. To restrict the execution of these tools only to the root and

the *studRoot*, the execution permission given to others, users other than the root and *studRoot* group, must be taken away. The following commands can therefore be issued:.

```
#chmod o-x mtrace
#chmod o-x mrinfo
```

Final permissions on both tools can be viewed with *ls -l* command.

```
#ls -l mtrace
-rwsr-xr--   1 root studRoot 123183 Jul 30 18:45 mtrace*
#ls -l mrinfo
-rwsr-xr--   1 root studRoot  91165 Jul 30 18:48 mrinfo*
```

With these mode bit settings, *mrinfo* and *mtrace* can be run only by the root and users belonging to the *studRoot* group.

APPENDIX E. SOURCE CODES FOR MULTICAST MONITORING TOOLS

A. AUTOMATED MROUTER CHECKING PROGRAM (AMCP)

```
#!/usr/local/bin/perl
#_____________________________________________________________________
#
# Program: MBmonitor.pl
#
# Usage:   MBmonitor.pl [mrouter information file]
#
# Purpose: MBmonitor.pl is used to determine the current status of mrouter(s)
#          and multicast routing daemon(s) (mrouteds) running on these
#          mrouter(s).
#
# Description: In order to determine the current status, MBmonitor runs ping
#              and mrinfo. Ping is run to determine if the mrouter is dead or
#              alive.Unless the packet loss is not 100%, the mrouter is
#              assumed to be alive. If the packet loss is 100%, the mrouter
#              is dead and the mrouted is not running. If the mrouter is
#              alive, MBmonitor.pl runs the mrinfo tool in order to determine
#              the current status of mrouted. Any response received from
#              mrouter indicates the mrouted is running on that mrouter. The
#              lack of response means that the mrouted is not running.
#              The first time mrouter or mrouted is detected as down, an
#              alert message is sent to the person who is responsible for
#              that mrouter and this status change is logged in
#              mrouter_satatus and mrouted_status directories under the
#              working directory. Mrouter status change is logged into a file
#              named <mroutername>.dead. Mrouted status change is logged into
#              a file named <mroutername>.mrouted.dead. Until the mrouter or
#              the mrouted is recovered this current status is logged into
#              these files. No other message is sent to the responsible
#              person unless the mrouted or mrouter is recovered. When the
#              mrouter or the mrouted is recovered, the log files are
#              renamed. "log" extension is added at the end of the log file.
#              MBmonitor.pl reads mrouter information from either a user
#              specified file or a default file located under the working
#              directory. MBmonitor.pl creates a Web page in html format in
#              order to display the current status of both mrouter and
#              mrouted as well as the other information specified in mrouter
#              information file.
#
$URL = 'http://www.stl.nps.navy.mil/~erdogan/mbone/report.html' ;
$SOURCE_URL = 'http//www.stl.nps.navy.mil/~erdogan/mbone/MBmonitor.pl.txt';
#
$working_directory = '/home/students/erdogan/.public_html/mbone';
#
# Author:     Ridvan Erdogan
#
# Revised:    8 Aug 1996
#_____________________________________________________________________
```

```
($info_file) = @ARG;  # get the file name
# look at the default file
$info_file = $working_directory.'/mrouter.info' unless $info_file;
$refresh_rate = 60 ;

open (TIME, "date|");  # get the current time
$date = <TIME>;

open (HTML, ">$working_directory/report.html");

print HTML <<HTML_HEAD; # create a report in HTML format

<HTML>
<HEAD>
<TITLE> Mrouter Report </TITLE>
</HEAD>

<BODY>

<META HTTP-EQUIV="Refresh" CONTENT=$refresh_rate>

<P>URL of this page is <I>$URL</I></P><P>
<CENTER>
This report was generated by AMCP on $date
<P>
<H3>Mrouter Status Report for NPS</H3>
<TABLE BORDER=5 CELLSPACING=0 CELLPADDING=5>
<TR>
    <TH ALIGN=LEFT><H3>Mrouter Host Name</H3>
    <TH ALIGN=LEFT><H3>Mrouter IP Address</H3>
    <TH ALIGN=LEFT><H3>Mrouter Physical Location</H3>
    <TH ALIGN=LEFT><H3>Status of Mrouter</H3>
    <TH ALIGN=LEFT><H3>Status of Mrouted</H3>
    <TH ALIGN=LEFT><H3>Point of Contact</H3>

HTML_HEAD

open (INFO, "<$info_file") ||
            die "Sorry cannot open file $info_file or file does not exist!";

while (<INFO>) {                # read the mrouter information
   next if ( /^#/ || /^$/ );   # ignore the commented lines
   # parse the line
   ($name , $ip_address, $location, $point_of_contact) = split;
   # send 2 ping messages to the mrouter
   open (PING, "ping -q -c 2 $ip_address|");

 CHECK: # check if the mrouter is alive or not
   while ( $status = <PING>){

      if ( $status =~ /packet/ ) {

         (@result) = split (' ', $status);
```

```
if ( $result[6] eq '100%' ) { # mrouter is dead (packet loss is %100)
   $mrouter_status = "not alive";
   $mrouted_status = "not running";

   if ( !( -e "$working_directory/mrouter_status/$name.down")) {

      # mrouter is detected as dead for the first time
      open (DOWN, ">$working_directory/mrouter_status/$name.down");
      print DOWN ("Host Name :        $name\n");
      print DOWN ("Host IP Address:  $ip_address\n");
      print DOWN ("\nTime:              $date");
      print DOWN ("Current Status:   $name is not alive and mrouted
                                     is not running\n");
      close (DOWN);

      &send_mail ($name, $ip_address, $current_status,
                                      $mrouted_status)
   } #end of if (!(-e ...))

   else { #mrouter is still down

      open (STILL_DOWN, ">>$working_directory/mrouter_status/
                        $name.down\n") || die "can't open the file";
      print STILL_DOWN ("Time:          $date");
      print STILL_DOWN ("Current Status: $name is still not alive and
                                    mrouted is still not running\n");
      close (STILL_DOWN);

   } #end of else
} #end of if ($result...)

else { #mrouter is alive

   $mrouter_status = "alive";

   if ( -e "$working_directory/mrouter_status/$name.down") {
      #if there is a log file, write the current status into it
      #and rename the file
      open (UP,">>$working_directory/mrouter_status/$name.down");
      print UP ("Time:          $date");
      print UP ("Current Status:  Mrouter is alive\n");
      close (UP);

      system ("mv $working_directory/mrouter_status/$name.down
                  $working_directory/mrouter_status/$name.down.log");

   } # end of if ( -e ...)
   open (MRINFO, "/usr/local/bin/mrinfo $ip_address|");

   if ( !<MRINFO> ) { # no response is gotten from the mrouter
     $mrouted_status = "not responding, possibly not running";

     if(-e "$working_directory/mrouted_status/$name.mrouted.down" ){
```

```
            #mrouted is still not responding
            open (MROUTED_DOWN, ">>$working_directory/mrouted_status/
                                  $name.mrouted.down\n") ;
            print MROUTED_DOWN ("\nTime :        $date");
            print MROUTED_DOWN ("Current Status: Mrouted is still
                                                 $mrouted_status");

            close (MROUTED_DOWN);

         } #end of if ( -e...)

         else { #mrouted is detected as dead for the first time

            open (M_DOWN ,">$working_directory/mrouted_status/
                           $name.mrouted.down");
            print M_DOWN ("Host Name:        $name\n");
            print M_DOWN ("Host IP Address : $ip_address\n");
            print M_DOWN ("\nTime :            $date");
            print M_DOWN ("Current Status:  Mrouted is
                                           $mrouted_status\n");
            close (M_DOWN);

           &send_mail ($name, $ip_address, $current_status,
                                              $mrouted_status);
         } #end of else

    } #end od if (!<MRINFO>)

    else {  #mrouted is running
       $mrouted_status = "running";

         if ( -e "$working_directory/mrouted_status/mrouted_status/
                  $name.mrouted.down"){
            open (M_UP,">>$working_directory/mrouted_status/
                          $name.mrouted.down");
            print M_UP ("\nTime:          $date");
            print M_UP ("Current Status:  Mrouted is running\n");

            system ("mv $working_directory/mrouted_status/$name.down
                        $working_directory/mrouted_status/
                        $name.mrouted.down.log");
            close (M_UP);
         } #end of if ( -e "$name...)
       } # end of else

         close (MRINFO);
   }

} #end of if ($status...)

else { #ignore all the lines that does not include "packet"
    next CHECK;
}
```

```
    } #end of  CHECK


print HTML <<FILL_TABLE;
<TR>
<TH ALIGN=LEFT>$name
<TH ALIGN=LEFT>$ip_address
<TH ALIGN=LEFT>$location
<TH ALIGN=LEFT> <B> $mrouter_status</B>
<TH ALIGN=LEFT> <B> $mrouted_status</B>
<TH ALIGN=LEFT> <A HREF="mailto:$point_of_contact">$point_of_contact</A>

FILL_TABLE


} #end of while (<INFO>)

print HTML<<HTML_BOTTOM;
</TABLE>
</CENTER>

<P>
<HR>

Point of contact:
<A HREF="mailto:erdogan\@cs.nps.navy.mil">erdogan\@cs.nps.navy.mil</a>
</BODY>
</HTML>

HTML_BOTTOM

close HTML;

exit;
#_____________________________________________________________________
# Name:         send_mail
#
# Purpose:      send_mail  is used to send a mail to the point of contact
#               specified in mrouter.info file
# Description: send_mail fills a form by using the information provided by
#               the caller, writes this form into a file and sends it to the
#               person who is responsible from that mrouter.
# Parameters:  $host_name - name of the mrouter
#               $host_ip_address  - ip address of the mrouter
#               $host_status  - current status of the mrouter (dead or alive)
#               $m_status  - current status of mrouted  (running or not
#               running)
#_____________________________________________________________________

sub send_mail {

local ($host_name, $host_ip_address, $host_status, $m_status) = @_ ;

open (REPORT, ">report.txt");
```

```
print REPORT <<END_OF_REPORT; # alert message to the point of contacts

 This is a report generated by Automated Mrouter Checking Program (AMCP)!

 The mrouted (multicast routing deamon) is normally running
 on host $host_name \($host_ip_address\) for your subnetwork.

 The ACMP  program detected on $date that $host_name is $host_status and
 mrouted is $m_status.

 Please check it and make mrouted running to restore Multicast Backbone
(MBone) connectivity.

 The current status of all mrouters on the NPS Campus can be found at
      http://blackand.stl.nps.navy.mil/~erdogan/mbone/report.html

 This page is updated at one hour intervals.  You will not receive further
 reports unless $host_name status changes.

 Further unicast connectivity status is available at
     http://www.stl.nps.navy.mil/~iirg/atm/monitoring/Ping_pages/NPS/
            status.html

 Thank you.

 P.S. For more information about these projects, please see the
      Information Infrastructure Research Group at
      http://www.stl.nps.navy.mil/~iirg
      or contact Don Brutzman brutzman\@nps.navy.mil

END_OF_REPORT

close REPORT;

system ("/usr/local/bin/elm -s Mrouter_is_dead $point_of_contact
                            < report.txt");

} #end of sub send_mail
```

B. MROUTER.INFO FILE

The AMCP reads the information related to mrouters by dafault from a file called *mrouter.info*. The user-specified file may also provided as a command line argument. The AMCP looks for this file under the directory specified in the program source. The following is the *mrouter.info* file used by AMCP running for the local NPS MBone. # indicates the commented lines. They are ignored by the AMCP.

```
#Host_Name                  Host_IP_Adress   Physical_Location  Point_of_Contact
#
mbone.nps.navy.mil          131.120.254.59   Computer_Center    romo@nps.navy.mil
mbone.cc.nps.navy.mil       131.120.53.21    Computer_Center    romo@nps.navy.mil
cadet.stl.nps.navy.mil      131.120.64.17    STL_Lab.           mcgredo@stl.nps.navy.mil
intruder.aa.nps.navy.mil    131.120.149.55   H-103              tony@nps.navy.mil
ntc.nps.navy.mil            131.120.57.3     Computer_Center    ingram@nps.navy.mil
131.120.141.100             131.120.141.100  Auditorium         blau@nps.navy.mil
noise.usw.nps.navy.mil      131.120.140.62   R-107              hudson@usw.nps.navy.mil
indigo1.me.nps.navy.mil     131.120.151.221  ME_Comp_Lab        marco@me.nps.navy.mil
chandra.ece.nps.navy.mil    131.120.20.39    SP-308             voigt@ece.nps.navy.mil
auvonyx.me.nps.navy.mil     131.120.7.112    Golf_Course        marco@lex.me.nps.navy.mil
betelgeuse.cs.nps.navy.mil  131.120.211.3    SP-500             whalen@cs.nps.navy.mil
zeta.nps.navy.mil           131.120.254.222  Computer_Center    romo@nps.navy.mil
utumno.barrnet.net          131.119.244.11   Stanford_University jhawk@bbnplanet.com
```

C. MRINFO GATEWAY

```
#!/usr/bin/perl
#_____________________________________________________________________
#
# Program: mrinfo_gw.cgi
#
# Usage:   mrinfo_gw.cgi
#
# Purpose: mrinfo_gw.cgi is used to get the configuration information from an
#          mrouter. It provides world-wide accessibility to mrinfo tool which
#          normally can be run only by super-user (root)
#
# Description: mrinfo_gw.cgi simply runs mrinfo. Since it is a CGI/Perl
#              script, it can be invoked via a Web page. It creates a
#              graphical user interface and makes use easier. It gets the
#              mrouter configuration information and displays the information
#              via a Web page. The Web page is created automatically by the
#              script. The mrouter IP address or the host name must be
#              provided by the user. If no information is provided, the user
#              is informed about that and asked for an IP number or a host
#              name.

$URL = 'http://www.stl.nps.navy.mil/~erdogan/mbone/report.html' ;
$SOURCE_URL = 'http//www.stl.nps.navy.mil/~erdogan/mbone/mrinfo_gw.cgi.txt';
#
# Author:     Ridvan Erdogan
#
# Revised:    8 Aug 1996
#_____________________________________________________________________

require "cgi-lib.pl";  # include the predefined subroutines

MAIN:
{
   if (! ReadParse (*preferences )) {  # script is invoked without providing
                                       # any parameters
      print &PrintHeader;
      print &HtmlTop;
      &PrintForm;      # create a GUI
      print &HtmlBot;
    }
    else {
      print &PrintHeader;
      &ProcessForm;
    }
```

```
#_____________________________________________________________________
# Name: PrintForm
# Purpose: printForm is used to create a graphical user interface
# Description: A graphical user interface is created and displayed on a Web
#              page in order to get the user inputs.
#_____________________________________________________________________

sub PrintForm {

printf <<END_OF_FORM;    # describe the GUI to the Web Browser
<center>

<h3>NPS Mrinfo Gateway</h3>

   <form method=get action="mrinfo_gw.cgi">
   <table border=5 cellspacing=0 cellpadding=5>
<tr>
   <td align=left>
   <img src="images/redball.gif"> You may want to take a look at
   <a href="mrinfo_man.html"  target="new">the man page of mrinfo</a>
   before you use this gateway!
<tr>
   <td align=left>
   Mrouter IP Address or Host name <input type=text name=mrouter size=15> <p>
<tr>
   <td align=center>
   <input type=submit value="Submit">
 </table>
 </form>
 </center>
END_OF_FORM

} #end of PrintForm
```

```
#_____________________________________________________________________
# Name: ProcessForm
# Purpose: ProcessForm is used to run the actual program according to the
information provided
#        by the user
# Description: ProcessForm gets the information provided by the user via the
GUI created by
#           PrintForm subroutine and runs the actual program
# Parameters:   $mrouter  - IP address or the name of the mrouter to be
queried
#_____________________________________________________________________

sub ProcessForm {

open (TIME, "date|");  # get the current time
$date = <TIME>;
$mrouter = $preferences{'mrouter'};  # get the IP number or the host name of
                                     # the mrouter

print <<HTML_HEAD;     #create a Web page to display the result
<HTML>
<HEAD>
<TITLE> Mrouter Info </TITLE>
</HEAD>
<BODY>
<P>
<CENTER>This report was generated on $date
<H2>Configuration Details for $mrouter</H2>
</CENTER>
HTML_HEAD
if ( $mrouter ) {  # check if the destination mrouter IP address or host name
                   # is provided

   open (MRINFO, "/usr/local/bin/mrinfo $mrouter|");
   $result = <MRINFO> ;

   if ( !$result ){  # no response is gotten from the mrouter
     print " $mrouter is not responding <br>";
     print " Either the mrouted is not running or the mrouter is behind a
             firewall!";
   }
   while ( $result ) {  # mrouter responded the query
      if (  ($result =~ /version/ ) || ($result =~ /querier/) ) {
          print "$result <br>";
      }
      else {  # create hot links to the tunneled mrouters

        (@data) = split (' ', $result);
         print "$data[0]  $data[1]  ";
         print "<a href=\"http://blackand.stl.nps.navy.mil/~erdogan/mbone/
                         mrinfo_gw.cgi?mrouter=$data[2]\"> $data[2]</a>";
         print "  $data[3] $data[4] <br>";
      }
```

```
      $result = <MRINFO>;

   } #end of while ( $result...)
} #end of if ( $mrouter )

else {   # no mrouter IP address or host name is provided
   print " You need to provide the name or the IP Address of the
           mrouter!<br>";
   print " Please try again<p>";
}

print <<HTML_BOTTOM;   # complete the html format
<P>
<HR>
Back to <A HREF="mrinfo_gw.cgi"> Mrinfo Gateway </A>
<P>URL : <I>$URL</I><BR>
Source code is available at $SOURCE_URL <P>
Point of contact : <A HREF="mailto:erdogan\@cs.nps.navy.mil">
                           erdogan\@cs.nps.navy.mil</a>
</BODY>
</HTML>
HTML_BOTTOM
exit;

} #end of ProcessForm
} #end of main
```

D. MTRACE GATEWAY

```
#!/usr/bin/perl
#_____________________________________________________________________
# Program: mtrace_gw.cgi
#
# Usage:   mtrace_gw.cgi
#
# Purpose: mtrace_gw.cgi is used to get the path taken by the multicast
#          packet from a source to destination via a multicast group address
#          and the statistical information along that path. It provides
#          world-wide accessibility to mtrace tool which is normally can be
#          run only by super-user (root)
#
# Description: mtrace_gw.cgi simply runs mtrace. Since it is a CGI/Perl
#              script, it can be invoked via a Web page. It creates a
#              graphical user interface and makes use easier. It gets the
#              route taken by the multicast packets and the statistical
#              information along that path and displays the information via a
#              Web page. The Web page is created automatically by the script.
#              The path is displayed hop-by-hop. The only required parameter
#              is the destination IP address or the host name. The default
#              source is the machine that mtrace_gw.cgi is running on. The
#              default multicast group is 224.2.0.1. The source IP address or
#              the host name and the multicast group address are optional
#              parameters. If any destination address is provided, the user
#              is asked for an IP number or a host name
#
$URL = 'http://www.stl.nps.navy.mil/~erdogan/mbone/report.html' ;
$SOURCE_URL = 'http//www.stl.nps.navy.mil/~erdogan/mbone/mtrace_gw.cgi.txt';
#
# Author:     Ridvan Erdogan
#
# Revised:    8 Aug 1996
#_____________________________________________________________________

require "cgi-lib.pl";  # include the predefined subroutines

MAIN:
{
   if (! ReadParse (*preferences )) {
      print &PrintHeader;
      print &HtmlTop;
      &PrintForm;  # create the GUI
      print &HtmlBot;
    }
    else {
      print &PrintHeader;
      &ProcessForm;   # run the actual program
    }
```

```
#_____________________________________________________________________
# Name: PrintForm
# Purpose: PrintForm is used to create a graphical user interface
# Description: A graphical user interface is created and displayed on a Web
#              page in order to get the user inputs.
#_____________________________________________________________________

sub PrintForm {

$default_receiver= $ENV {'SERVER_NAME'}; # machine that mtrace_gw.cgi is
                                        # running on
printf <<END_OF_FORM;
<center>

<h3>NPS Mtrace Gateway</h3>

   <form method=get action="mtrace_gw.cgi">
   <table border=5 cellspacing=0 cellpadding=5>
<tr>
   <td align=left>
   <img src="images/redball.gif"> You may want to take a look at
   <a href="mtrace_man.html"  target="new">the man page of mtrace</a>
   before you use this gateway!
<tr>
   <td align=left>
   Mtrace - <select name=switch>
            <option>l
            <option>M
            <option selected>s
            </select>  <input type=text name=source  size=15>
   to <input type=text name=destination value="$default_receiver" size=15><br>
<center> via multicast group
   <input type=text name=multicast_group value="224.2.0.1" size=20>
</center><p>
<tr>
   <td align=center>
   <input type=submit value="Submit">
 </table>
 </form>
 </center>
 END_OF_FORM

} #end of PrintForm
```

```
#_____________________________________________________________________
# Name: ProcessForm
# Purpose: ProcessForm is used to run the actual program according to the
#          information provided by the user
# Description: ProcessForm gets the information provided by the user via the
#              GUI created by PrintForm subroutine and runs the actual program
#Parameters:  $destination
#             $source
#             $multicast_group
#             $switch    - switches for mtrace
#_____________________________________________________________________

sub ProcessForm {

open (TIME, "date|");  # get the current time
$date = <TIME>;

#get the preferences
$destination = $preferences{'destination'};
$source = $preferences{'source'};
$multicast_group = $preferences{'multicast_group'};
$switch = $preferences{'switch'};

print <<HTML_HEAD;
<HTML>
<HEAD>
<TITLE> Mtrace Result </TITLE>
</HEAD>
<BODY BGCOLOR="#ffffff">
<P>
<CENTER>
This report was generated on $date
</CENTER>
HTML_HEAD

if ( $destination  ) { # check if the destination IP address or the host name
                       # is provided
  print "<center><H2>Multicast route from $source back to $destination </H2>
         </center>";
  open (MTRACE, "/usr/local/bin/mtrace -$switch $source $destination
                                              $multicast_group|");
  print "<pre>";
  while ( $result = <MTRACE>) {  # display the information gathered by the
                                 # mtrace
    print "$result<br>";
  } #end of while ( $result...)

  print "</pre>";
}

else {  # no IP address or host name is provided
  print "You need to type a destination Host Name or IP Number<p>";
  print "Please try again!";
```

```
}
print <<HTML_BOTTOM;  #complete the html format
</H2>
<P>
<HR>
Back to <A HREF="mtrace_gw.cgi"> Mtrace Gateway </A>
<P>URL : <I>$URL</I><BR>
Source code is available at $SOURCE_URL <P>
Point of contact : <A HREF="mailto:erdogan\@cs.nps.navy.mil">
                           erdogan\@cs.nps.navy.mil</a>
</BODY>
</HTML>
HTML_BOTTOM
exit;
} #end of ProcessForm
} #end of main
```

APPENDIX F. MRINFO MAN PAGE

Appendix F is the *man page* of *mrinfo*. Man pages (so called because they are designed for use with the `man` command) are one of two types of documentation that comes with UNIX systems. They are concise descriptions of individual commands, file formats, or library routines. They are usually kept on-line. (Nemeth, 1995) The following is the on-line information provided by the UNIX system when `man mrinfo` command is issued.

MRINFO(8) **UNIX System V** **MRINFO(8)**

NAME

mrinfo - Displays configuration info from a multicast router

SYNOPSIS

/usr/sbin/mrinfo [-d debug_level] [-r retry_count] [-t timeout_count] multicast_router

DESCRIPTION

mrinfo attempts to display the configuration information from the multicast router multicast_router. **mrinfo** uses the ASK_NEIGHBORS IGMP message to the specified multicast router. If this multicast router responds, the version number and a list of their neighboring multicast router addresses is part of that response. If the responding router has a recent multicast version number, then *mrinfo* requests additional information such as metrics, thresholds, and flags from the multicast router. Once the specified multicast router responds, the configuration is displayed to the standard output.

INVOCATION

"-d" option sets the debug level. When the debug level is greater than the default value of 0, addition debugging messages are printed. Regardless of the debug level, an error condition, will always write an error message and will cause *mrinfo* to terminate. Non-zero debug levels have the following effects:

level 1

packet warnings are printed to stderr.

level 2

all level 1 messages plus notifications down networks are printed to stderr.

level 3

all level 2 messages plus notifications of all packet timeouts are printed to stderr.

"-r retry_count" sets the neighbor query retry limit.

Default is 3 retry.

"-t timeout_count" sets the number of seconds to wait for a neighbor query reply. Default timeout is 4 seconds.

SAMPLE OUTPUT

```
mrinfo mbone.phony.dom.net
127.148.176.10 (mbone.phony.dom.net) [version 3.3]:
127.148.176.10 -> 0.0.0.0 (?) [1/1/querier]
127.148.176.10 -> 127.0.8.4 (mbone2.phony.dom.net) [1/45/tunnel]
127.148.176.10 -> 105.1.41.9 (momoney.com) [1/32/tunnel/down]
127.148.176.10 -> 143.192.152.119 (mbone.dipu.edu) [1/32/tunnel]
```

For each neighbor of the queried multicast router, the IP of the queried router is displayed, followed by the IP and name of the neighbor. In square brackets the metric (cost of connection), the threshold (multicast ttl) is displayed. If the queried multicast router has a newer version number, the type (tunnel, srcrt) and status (disabled, down) of the connection is displayed.

IMPORTANT NOTE

mrinfo must be run as root.

SEE ALSO

mrouted(8), map-mbone(8), mtrace(8)

AUTHOR

Van Jacobson

APPENDIX G. MTRACE MAN PAGE

Appendix G is the *man page* of *mtrace*. Man pages (so called because they are designed for use with the man command) are one of two types of documentation that comes with UNIX systems. They are concise descriptions of individual commands, file formats, or library routines. They are usually kept on-line. (Nemeth, 1995) The following is the on-line information provided by the UNIX system when `man mtrace` command is issued.

NAME

mtrace - print multicast path from a source to a receiver

SYNOPSIS

```
mtrace [ -g gateway ] [ -i if_addr ] [ -l ] [ -M ] [ -m
max_hops ] [ -n ] [ -p ] [ -q nqueries ] [ -r resp_dest ] [
-s ] [ -S stat_int ] [ -t ttl ] [ -v ] [ -w waittime ]
source [ receiver ] [ group ]
```

DESCRIPTION

Assessing problems in the distribution of IP multicast traffic can be difficult. *mtrace* utilizes a tracing feature implemented in multicast routers (mrouted version 3.3 and later) that is accessed via an extension to the IGMP protocol. A trace query is passed hop-by-hop along the reverse path from the receiver to the source, collecting hop addresses, packet counts, and routing error conditions along the path, and then the response is returned to the requestor. The only required parameter is the source host name or address. The default receiver is the host running *mtrace*, and the default group is "MBone Audio" (224.2.0.1), which is sufficient if packet loss statistics for a particular multicast group are not needed. These two optional parameters may be specified to test the path to some other receiver in a particular group, subject to some constraints as detailed below. The two parameters can be distinguished because the receiver is a unicast address and the group is a multicast address.

NOTE: For Solaris 2.4/2.5, if the multicast interface is not the default interface, the -i option must be used to set the local address.

OPTIONS

-g gwy Send the trace query via unicast directly to the multicast router gwy rather than multicasting the query. This must be the last-hop

router on the path from the intended source to the receiver.

CAUTION!! Versions 3.3 and 3.5 of mrouted will crash if a trace query is received via a unicast packet and mrouted has no route for the source address. Therefore, do not use the -g option unless the target mrouted has been verified to be 3.4 or newer than 3.5.

-i addr Use addr as the local interface address (on a multi-homed host) for sending the trace query and as the default for the receiver and the response destination.

-l Loop indefinitely printing packet rate and loss statistics for the multicast path every 10 seconds (see -S stat_int).

-M Always send the response using multicast rather than attempting unicast first.

-m n Set to n the maximum number of hops that will be traced from the receiver back toward the source. The default is 32 hops (infinity for the DVMRP routing protocol).

-n Print hop addresses numerically rather than symbolically and numerically (saves a nameserver address-to-name lookup for each router found on the path).

-q n Set the maximum number of query attempts for any hop to n. The default is 3.

-p Listen passively for multicast responses from traces initiated by others. This works best when run on a multicast router.

-r host Send the trace response to host rather than to the host on which *mtrace* is being run, or to a multicast address other than the one registered for this purpose (224.0.1.32).

-s Print a short form output including only the multicast path and not the packet rate and loss statistics.

-S n Change the interval between statistics gathering traces to n seconds (default 10 seconds).

-t ttl Set the ttl (time-to-live, or number of hops) for multicast trace queries and responses. The default is 64, except for local queries to the "all routers" multicast group which use ttl 1.

-v Verbose mode; show hop times on the initial trace and statistics display.

-w n Set the time to wait for a trace response to n seconds (default 3

seconds).

USAGE

How It Works

The technique used by the traceroute tool to trace unicast network paths will not work for IP multicast because ICMP responses are specifically forbidden for multicast traffic. Instead, a tracing feature has been built into the multicast routers. This technique has the advantage that additional information about packet rates and losses can be accumulated while the number of packets sent is minimized. Since multicast uses reverse path forwarding, the trace is run backwards from the receiver to the source. A trace query packet is sent to the last hop multicast router (the leaf router for the desired receiver address). The last hop router builds a trace response packet, fills in a report for its hop, and forwards the trace packet using unicast to the router it believes is the previous hop for packets originating from the specified source. Each router along the path adds its report and forwards the packet. When the trace response packet reaches the first hop router (the router that is directly connected to the source's net), that router sends the completed response to the response destination address specified in the trace query. If some multicast router along the path does not implement the multicast traceroute feature or if there is some outage, then no response will be returned. To solve this problem, the trace query includes a maximum hop count field to limit the number of hops traced before the response is returned. That allows a partial path to be traced. The reports inserted by each router contain not only the address of the hop, but also the ttl required to forward and some flags to indicate routing errors, plus counts of the total number of packets on the incoming and outgoing interfaces and those forwarded for the specified group. Taking differences in these counts for two traces separated in time and comparing the output packet counts from one hop with the input packet counts of the next hop allows the calculation of packet rate and packet loss statistics for each hop to isolate congestion problems.

Finding the Last-Hop Router

The trace query must be sent to the multicast router which is the last hop on the path from the source to the receiver. If the receiver is on the local subnet (as determined using the subnet mask), then the default method is to

multicast the trace query to all-routers.mcast.net (224.0.0.2) with a ttl of 1. Otherwise, the trace query is multicast to the group address since the last hop router will be a member of that group if the receiver is. Therefore it is necessary to specify a group that the intended receiver has joined. This multicast is sent with a default ttl of 64, which may not be sufficient for all cases (changed with the -t option). If the last hop router is known, it may also be addressed directly using the -g option). Alternatively, if it is desired to trace a group that the receiver has not joined, but it is known that the last-hop router is a member of another group, the -g option may also be used to specify a different multicast address for the trace query. When tracing from a multihomed host or router, the default receiver address may not be the desired interface for the path from the source. In that case, the desired interface should be specified explicitly as the receiver.

Directing the Response

By default, *mtrace* first attempts to trace the full reverse path, unless the number of hops to trace is explicitly set with the -m option. If there is no response within a 3 second timeout interval (changed with the -w option), a "*" is printed and the probing switches to hop-by-hop mode. Trace queries are issued starting with a maximum hop count of one and increasing by one until the full path is traced or no response is received. At each hop, multiple probes are sent (default is three, changed with -q option). The first half of the attempts (default is one) are made with the unicast address of the host running *mtrace* as the destination for the response. Since the unicast route may be blocked, the remainder of attempts request that the response be multicast to mtrace.mcast.net (224.0.1.32) with the ttl set to 32 more than what's needed to pass the thresholds seen so far along the path to the receiver. For the last quarter of the attempts (default is one), the ttl is increased by another 32 each time up to a maximum of 192. Alternatively, the ttl may be set explicitly with the -t option and/or the initial unicast attempts can be forced to use multicast instead with the -M option. For each attempt, if no response is received within the timeout, a "*" is printed. After the specified number of attempts have failed, *mtrace* will try to query the next hop router with a DVMRP_ASK_NEIGHBORS2 request (as used by the *mrinfo* program) to see what kind of router it is.

EXAMPLES

The output of *mtrace* is in two sections. The first section is a short listing of the hops in the order they are queried, that is, in the reverse of the order from the source to the receiver. For each hop, a line is printed showing the hop number (counted negatively to indicate that this is the reverse path); the multicast routing protocol (DVMRP, MOSPF, PIM, etc.); the threshold required to forward data (to the previous hop in the listing as indicated by the up-arrow character); and the cumulative delay for the query to reach that hop (valid only if the clocks are synchronized). This first section ends with a line showing the round-trip time which measures the interval from when the query is issued until the response is received, both derived from the local system clock. A sample use and output might be:

```
oak.isi.edu 80# mtrace -l caraway.lcs.mit.edu 224.2.0.3
Mtrace from 18.26.0.170 to 128.9.160.100 via group 224.2.0.3
Querying full reverse path...
0  oak.isi.edu (128.9.160.100)
-1  cub.isi.edu (128.9.160.153)  DVMRP  thresh^ 1  3 ms
-2  la.dart.net (140.173.128.1)  DVMRP  thresh^ 1  14 ms
-3  dc.dart.net (140.173.64.1)  DVMRP  thresh^ 1  50 ms
-4  bbn.dart.net (140.173.32.1)  DVMRP  thresh^ 1  63 ms
-5  mit.dart.net (140.173.48.2)  DVMRP  thresh^ 1  71 ms
-6  caraway.lcs.mit.edu (18.26.0.170)
Round trip time 124 ms
```

The second section provides a pictorial view of the path in the forward direction with data flow indicated by arrows pointing downward and the query path indicated by arrows pointing upward. For each hop, both the entry and exit addresses of the router are shown if different, along with the initial ttl required on the packet in order to be forwarded at this hop and the propagation delay across the hop assuming that the routers at both ends have synchronized clocks. The right half of this section is composed of several columns of statistics in two groups. Within each group, the columns are the number of packets lost, the number of packets sent, the percentage lost, and the average packet rate at each hop. These statistics are calculated from differences between traces and from hop to hop as explained above. The first group shows the statistics for all traffic flowing out the interface at one hop and in the interface at the next hop. The second group shows the

statistics only for traffic forwarded from the specified source to the specified group. These statistics are shown on one or two lines for each hop. Without any options, this second section of the output is printed only once, approximately 10 seconds after the initial trace. One line is shown for each hop showing the statistics over that 10-second period. If the -l option is given, the second section is repeated every 10 seconds and two lines are shown for each hop. The first line shows the statistics for the last 10 seconds, and the second line shows the cumulative statistics over the period since the initial trace, which is 101 seconds in the example below. The second section of the output is omitted if the -s option is set.

```
Waiting to accumulate statistics... Results after 101 seconds:
Source        Response Dest  Packet Statistics For  Only For Traffic
18.26.0.170     128.9.160.100  All Multicast Traffic  From 18.26.0.170
      |        __/ rtt  125 ms  Lost/Sent = Pct  Rate     To 224.2.0.3
      v      /  hop   65 ms  ----------------------- ----------------------
       18.26.0.144
       140.173.48.2    mit.dart.net
           |     ^     ttl    1       0/6    = --%   0 pps   0/2  = --%  0 pps
           v     |     hop    8 ms    1/52   =  2%   0 pps   0/18 =  0%  0 pps
       140.173.48.1
       140.173.32.1    bbn.dart.net
           |     ^     ttl    2       0/6    = --%   0 pps   0/2  = --%  0 pps
           v     |     hop   12 ms    1/52   =  2%   0 pps   0/18 =  0%  0 pps
       140.173.32.2
       140.173.64.1    dc.dart.net
           |     ^     ttl    3       0/271  =  0%  27 pps   0/2  = --%  0 pps
           v     |     hop   34 ms   -1/2652 =  0%  26 pps   0/18 =  0%  0 pps
       140.173.64.2
       140.173.128.1   la.dart.net
           |     ^     ttl    4      -2/831  =  0%  83 pps   0/2  = --%  0 pps
           v     |     hop   11 ms   -3/8072 =  0%  79 pps   0/18 =  0%  0 pps
       140.173.128.2
       128.9.160.153   cub.isi.edu
           |      \__  ttl    5         833          83 pps     2          0 pps
           v         \ hop   -8 ms      8075         79 pps     18         0 pps
       128.9.160.100   128.9.160.100
```

```
        Receiver      Query Source
```

Because the packet counts may be changing as the trace query is propagating, there may be small errors (off by 1 or 2) in these statistics. However, those errors should not accumulate, so the cumulative statistics line should increase in accuracy as a new trace is run every 10 seconds. There are two sources of larger errors, both of which show up as negative losses:

- If the input to a node is from a multi-access network with more than one other node attached, then the input count will be (close to) the sum of the output counts from all the attached nodes, but the output count from the previous hop on the traced path will be only part of that. Hence the output count minus the input count will be negative.
- In release 3.3 of the DVMRP multicast forwarding software for SunOS and other systems, a multicast packet generated on a router will be counted as having come in an interface even though it did not. This creates the negative loss that can be seen in the example above. Note that these negative losses may mask positive losses. In the example, there is also one negative hop time. This simply indicates a lack of synchronization between the system clocks across that hop. This example also illustrates how the percentage loss is shown as two dashes when the number of packets sent is less than 10 because the percentage would not be statistically valid. A second example shows a trace to a receiver that is not local; the query is sent to the last-hop router with the -g option. In this example, the trace of the full reverse path resulted in no response because there was a node running an old version of mrouted that did not implement the multicast traceroute function, so *mtrace* switched to hop-by-hop mode. The Route pruned error code indicates that traffic for group 224.2.143.24 would not be forwarded.

```
oak.isi.edu 108# mtrace -g 140.173.48.2 204.62.246.73 \
                                butter.lcs.mit.edu 224.2.143.24
Mtrace from 204.62.246.73 to 18.26.0.151 via group 224.2.143.24
Querying full reverse path... * switching to hop-by-hop:
0   butter.lcs.mit.edu (18.26.0.151)
-1  jam.lcs.mit.edu (18.26.0.144)  DVMRP  thresh^ 1  33 ms  Route pruned
-2  bbn.dart.net (140.173.48.1)  DVMRP  thresh^ 1  36 ms
-3  dc.dart.net (140.173.32.2)  DVMRP  thresh^ 1  44 ms
```

```
-4  darpa.dart.net (140.173.240.2)  DVMRP  thresh^ 16  47 ms
-5  * * * noc.hpc.org (192.187.8.2) [mrouted 2.2] didn't respond
Round trip time 95 ms
```

AUTHOR

Implemented by Steve Casner based on an initial prototype written by Ajit Thyagarajan. The multicast traceroute mechanism was designed by Van Jacobson with help from Steve Casner, Steve Deering, Dino Farinacci, and Deb Agrawal; it was implemented in mrouted by Ajit Thyagarajan and Bill Fenner. The option syntax and the output format of *mtrace* are modeled after the unicast traceroute program written by Van Jacobson.

LIST OF REFERENCES

Aarnio, M., "MBone over Frame Relay," electronic-mail correspondence to Ltjg. Ridvan Erdogan, 16 February 1996.

Bigelow, R. J., *Internetworking: Planning and Implementing a Wide-Area Network (WAN) for K-12 Schools*, Master's Thesis, Naval Postgraduate School, Monterey, California, September 1995. Available at
http://www.stl.nps.navy.mil/~rjbigelo/thesis/toc.html Hypertext Version
http://www.stl.nps.navy.mil/~rjbigelo/thesis/psversion.html Postscript Version
http://www.stl.nps.navy.mil/~rjbigelo/thesis/textonly.txt Text Version

Braudes, R. and Zabele, S., *Requirements for Multicast Protocols*, IETF Network Working Group Request for Comments, RFC 1458, May 1993. Available at: *http://ds.internic.net/rfc/rfc1458.txt*

Brenner, S.E., Aoki, E., *Introduction to CGI/Perl,* M&T Books, New York, 1996.

Brown, C., Bradley, T. and Malis, A., *Multiprotocol Interconnect over Frame Relay,* IETF Network Working Group Request For Comments: 1490 (RFC 1490), July, 1993. Available at *ftp://ds.internic.net/rfc/rfc1490.txt*

Brown, C. and Malis, A., *Multiprotocol Interconnect over Frame Relay,* IETF Network Working Group Internet Draft, July 10, 1996. Available at
ftp://ds.internic.net/internet-drafts/draft-ietf-ion-fr-update-01.txt

Casner, S., "Are You on the MBone?," *IEEE Multimedia,* vol.1 no. 2, pp. 76-79, Summer 1994.

Cisco Systems Inc., WWW Home Page, San Jose, California, 1996. Available at
http://www.cisco.com

Cisco System Inc., *UniverCD*, vol. 3, no.9, Cisco Systems Knowledge Products, San Jose, California, 1996.

Comer, D. E., *Internetworking with TCP/IP*, Volume I, Prentice-Hall, Englowood Cliffs, N.J., 1991.

Courtney, D. M., *Internetworking: ATM LAN at NPS,* Master's Thesis, Naval Postgraduate School, Monterey, California, September 1996. Available at
http://vislab-www.nps.navy.mil/~iirg/courtney/thesis.html

Deering, S.E., *Host Extensions for IP Multicasting,* IETF Network Working Group Request for Comments (RFC 1112), August 1989. Available at *ftp://ds.internic.net/rfc/rfc1112.txt*

Deering, S.E. and Cheriton, D.R., "Multicast Routing in Datagram Internetworks and Extended LANs," *ACM Transaction on Computer Systems*, vol. 8, no. 2, pp. 85-110, May 1990.

Deering, S.E., *Multicast Routing in a Datagram Internetwork*, the Doctoral Dissertation, Stanford University, 1991. Available at:
ftp://gregorio.stanford.edu/vmtp-ip/sdthesis.part1.ps.Z Part 1
ftp://gregorio.stanford.edu/vmtp-ip/sdthesis.part2.ps.Z Part 2
ftp://gregorio.stanford.edu/vmtp-ip/sdthesis.part3.ps.Z Part 3

Deering, S., Estrin, D., Farinacci, D., Jacobson, V., Liu, C., Wei, L., Sharma, P. and Helmy, A., *Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification,* IETF Interdomain Multicast Routing (IDMR) Working Group Internet Draft, June 6, 1996. Available at *ftp://ds.internic.net/internet-drafts/draft-ietf-idmr-PIM-SM-spec-05.ps*

Edwards, E., *Internetworking: Automated Local and Global Network Monitoring* , Master's Thesis, Naval Postgraduate School, Monterey, California, September 1996. Available at *http://www.stl.nps.navy.mil/~iirg/edwards/thesis.html*

Emswiler, T., *Internetworking: Worldwide Multicast of the Hamming Lectures for Distance Learning,* Master's Thesis, Naval Postgraduate School, Monterey California, September 1995.

Ericson, H., "MBONE: The Multicast Backbone," *Communications of the ACM,* vol. 37, pp. 54-60, August 1994.

Fenner, W., *Internet Group Management Protocol, Version 2,* IETF Inter-domain Multicast Routing (IDMR) Working Group Internet Draft, May 20, 1996. Available at *ftp://ds.internic.net/internet-drafts/draft-ietf-idmr-igmp-v2-03.txt*

Frame Relay Forum, *Frame Relay: Networks for Tomorrow and Today,* Frame Relay Forum Publications, 1994. Available at: *http://frame-relay.indiana.edu/4000/4000index.html*

Henderson, F., McCoy, J., "Less is Faster", *Local-Area Network Magazine,* vol. 6, no. 7, p. 48, 1991.

Jacobson, V., Estrin, D., Farinacci, D., Liu C., Wei, L., Sharma, P. and Helmy, A., *Protocol Independent Multicast (PIM), Dense Mode Protocol Specofication,* IETF Inter Domain Muticast Routing (IDMR) Working Group Draft, January 17, 1996. Available at: *ftp://ds.internic.net/internet-drafts/draft-ietf-idmr-pim-dm-spec-03.ps*

Kumar, V., *MBone: Interactive Multimedia on the Internet*, New Riders Publishing, Indianapolis, Indiana, 1996

Matray, K., "Destination Tomorrow: Making Connections for the Ultimate field Trip" *CalREN grant proposal*, Monterey Unified School District (MPUSD), Monterey, California, 14 Mar 1994. Available at *http://www.pacbell.com/cgi-bin/textit?/SuoerHi/CalREN/Projects/educatio-9.html*

Macedonia, M.R. and Brutzman, D.P., "MBone Provides Live Audio and Video Across the Internet," *IEEE Computer,* vol. 27, no. 4, pp. 30-36, April 1994. Available at *ftp://taurus.cs.nps.navy.mil/pub/i3la/mbone.ps* Postscript Form

ftp://taurus.cs.nps.navy.mil/pub/i3la/mbone.txt Text Form
ftp://taurus.cs.nps.navy.mil/pub/i3la/mbone.html Hypertext Form

Microsoft Corporation, *Microsoft Windows 95 Reseource Kit*, Microsoft Press, Redmond, Washington, 1995.

Nemeth, E., Snyder, G.,Seebass, S. and Hein, T.R., *Unix System Administration Handbook*, Second Edition, Prentice Hall, Inc., Upper Saddle River N.J, 1995.

PacBell, *Frame Relay User's Guide*, 1994.

Partridge, C., *Gigabit Networking*, Addison-Wesley Publishing Co., Massachusetts, 1994.

Partridge, C., Waitzman, D. and Deering, S., *Distance Vector Multicast Routing Protocol*, IETF Network Working Group Request for Comments: 1075 (RFC1075), November, 1988. Available at *ftp://ds.internic.net/rfc/rfc1075.txt*

Reynolds, J., Postel, J., *Assigned Numbers*, IETF Network Working Group Request for Comments: 1700 (RFC1700). Available at *ftp://ds.internic.net/rfc/rfc1700.txt*

Savetz, K., Randall, N. and Lepage, Y., *MBone: Multicasting Tomorrow's Internet*, IDG Books Worldwide Inc., Foster City, California, 1996.

Semeria, C. and Maufer, T., *Introduction to IP Multicast Routing*, Internet Draft, March 1996. Available at *ftp://ds.internic.net/internet-draft/draft-rfced-info-semeria-00.txt*

Stallings, W., *Data and Computer Communications*, McMillan Publishing Co., Englewood Cliffs N.J., 1991.

Stallings, W., *ISDN and Broadband ISDN with Frame Relay and ATM*, Prentice-Hall, Inc., Englewood Cliffs N.J., 1995.

Stevens, W.R., *TCP/IP Illustrated: The Protocols, Volume 1*, Addisson-Wesley Publishing Co. Inc., Reading Massachachusetts, 1994.

Swallow, G., *Frame Relay PVC Multicast Service and Protocol Description*, Implementation Agreement, Frame Relay Forum, October 21, 1994. Available at: *http://frame-relay.indiana.edu/5000/Approved/FRF.7/FRF7-TOC.html*

Tamer, M., *Internetworking: Multicast and ATM Network Prerequisites for Distance Learning*, Master's Thesis, Naval Postgraduate School, Monterey, California, September 1996. Available at *http://www.stl.nps.navy.mil/~iirg/tamer/thesis.html*

Wall, L. and Schwartz, R.L., *Programming Perl*, O'Reilly & Associates, Inc., Sebastapol, California, 1991. Information is available at *http://www.ora.com/catalog/pperl*

INITIAL DISTRIBUTION LIST

No. of Copies

1. Defense Technical Information Center 2
   8725 John J. Kingman Road., STE 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, California 93943-5101

3. Dr. Don Brutzman, Code UW/Br 4
   Naval Postgraduate School
   Monterey, California 93943-5101

4. Dr. Jim Eagle, Chair, Code UW 1
   Naval Postgraduate School
   Monterey, California 93943-5101

5. Richard S. Elster, Ph.D 1
   Provost & Academic Dean, Code 01
   Naval Postgraduate School
   Monterey, California 93943-5101

6. Dr. James C. Emery, Code 05 1
   Dean of Computing
   Naval Postgraduate School
   Monterey, California 93943-5101

7. Dr. Ted Lewis, Chair, Code CS 1
   Naval Postgraduate School
   Monterey, California 93943-5101

8. Don McGregor, Code C3 1
   Naval Postgraduate School
   Monterey, California 93943-5101

9. Dave Norman, Code 51 1
   Director, W.R. Church Computer Center
   Naval Postgraduate School
   Monterey, California, 93943-5101

10. Gary Porter, Code C3 1
Naval Postgraduate School
Monterey, California 93943-5101

11. Dr. Maxine Reneker, Code 52 1
Director, Dudley Knox Library
Naval postgraduate School
Monterey, California 93943-5101

12. Maurice D. Weir 1
Associate Provost For Instruction, Code 01B
Naval Postgraduate School
Monterey, California 93943-5101

13. Dr. Michael J. Zyda, Code CS/Zk 1
Naval Postgraduate School
Monterey, California 93943-5101

14. Roland Baker 1
Santa Cruz County Office of Education
Media and Technology Services
809 Bay Avenue
Capitola, California 95010

15. Dr. Carl R. Berman, Jr 1
Office of the Provost
CSU Monterey Bay
100 Campus Center
Seaside CA 93955-8001

16. Jeff Bryant 1
Monterey Bay Aquarium
886 Cannery Row
Monterey, California, 93940

17. Jon Crowcroft 1
Depatrment of Computer Science
University College London
Gower Street
London WC1E 6BT United Kingdom

18. Steve Deering 1
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, California 94304

19. Deniz Harp Okulu Komutanligi 1
81704 Tuzla, Istanbul, TURKEY

20. Deniz Kuvvetleri Komutanligi 1
Personel Daire Baskanligi
Bakanliklar, Ankara, TURKEY

21. Ltjg. Ridvan Erdogan, TUN 1
Ataturk Mahallesi Ogretmenler Caddesi 10/4
35750, Odemis, Izmir, TURKEY

22. Dino Farinacci 1
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706

23. Bill Fenner 1
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, California 94304

24. Dr. Nancy Giberson 1
Santa Cruz County Office of Education
809 Bay Avenue
Capitola, California 95010

25. Bruce Gritton 1
Monterey Bay Aquarium Research Institute (MBARI)
160 Central Ave
Pacific Grove, California, 93950

26. Robert Gwinn 1
Instructional Leader
Main Street Middle School
441 Main Street
Soledad, California

27. Mark Handley 1
Depatrment of Computer Science
University College London
Gower Street
London WC1E 6BT United Kingdom

28. John Hawkinson 1
150 CambridgePark Drive
Cambridge, MA 02140

29. Dr. G. Ross Heath 1
Executive Director
Monterey Bay Aquarium Research Institute (MBARI)
P.O. Box 628
Moss Landing, California, 95039-0628

30. Birt Johnson 1
Pacific Bell
340 Pajero Street, Room 131
Salinas, California, 93901

31. Vinay Kumar 1
New Riders Publishing
210 West 103rd Street
Indianapolis, IN 46290

32. Yves Lepage 1
IDG Books Worldwide Inc.
Foster City, California 94404

33. Syd Leung 1
Pacific Bell
2600 Camino Ramon, Room 35306
San Ramon, California 94105

34. Brian Lloyd 1
Llyod Internetworking
3461 Robin Lane
Cameron Park, California 95681

35. Michael McCann 1
Monterey Bay Aquarium Research Institute
P.O. Box 628
Moss Landing, California, 95039-0628

36. Steve McCanne 1
Room 633 Soda Hall #1776
University of California
Berkeley, California 94720-1776

37. Dr. Ray McLain 1
Moss Landing Marine Laboratories
P.O. Box 450
Moss Landing, California, 95039

38. Dr. Michael R. Macedonia 1
Fraunhofer Center for Research in Computer Graphics, Inc.
Vice-President
167 Angel Street
Province, RI 02906

39. Dr. Pat Mantey 1
Chair, Computer Engineering
University of California, Santa Cruz
Santa Cruz, California 95064

40. Kam Matray 1
Monterey Bay Technology Education Center
Monterey Peninsula Unified School District
P.O. Box 1031
Monterey, California 93942-1031

41. Mike Mellon 1
Monterey County Office of Education
Instructional Resources and Technology
PO Box 80851
Salinas, California 93912-0851

42. Neil Randall 1
IDG Books Worldwide Inc.
Foster City. California 94404

43. Deborah Richards 1
Industry Consultant, Pacific Bell
2460 North Main Street
Salinas, California 93906

44. Kevin Savetz 1
IDG Books Worldwide Inc.
Foster City. California 94404

45. Diane Siri 1
County Superintendent of Schools
Santa Cruz County Office of Education
809 Bay Avenue
Capitola, California 95010

46. Dr. Fred Siff, Associate Vice Chancellor 1
Communications and Technology Services
University of California, Santa Cruz
Santa Cruz, California 95064

47. David Stihler 1
Network Administrator
Monterey County Office of Education
901 Blanco Circle
P.O. Box 80851
Salinas, California 93905-0851

48. LTJG Murat Tamer, TUN 1
Cemenzar Dr. Fazil Gokce Oren sk.
Tac apt. No 29/6
81080 Goztepe, Istanbul, TURKEY

49. Chris Taylor 1
Director of Computing and Computer Resources (CCR)
California State University, Monterey Bay
100 Campus Center
Seaside, California 93955-8001

50. Ajit Thyagarajan 1
121 Evans Hall
Department of Electrical Engineering
University of Delaware
Newark, DE 19716

51. Jim Warner 1
Network Engineer, University of California Santa Cruz
CATS/Network & Telco Services
11 Communications Bldg.
Santa Cruz, California 95064

52. Dr. Steve Webster 1
Director of Education
Monterey Bay Aquarium
886 Cannery Row
Monterey, California 93940

53. Victoria Welborn 1
Head Science Library
University of California
Santa Cruz, California 95064